MASTER EN SCIENCES INFORMATIQUES

L'administration des réseaux
concepts et outils

Tollet, Bernard

Award date:
1998

Facultés Universitaires Notre-Dame de la Paix, Namur

Institut d'Informatique

L'Administration des Réseaux
Concepts et Outils

Bernard TOLLET

Mémoire présenté en vue
de l'obtention du grade de
Licencié et Maître en Informatique

Année académique 1997-1998

TABLE DES MATIERES

PREFACE

Si nous remontons cinquante ans en arrière, on se rend compte que l'informatique n'était encore qu'à ses balbutiements. Elle n'était utilisée que pour des calculs complexes ou des tâches simples par les universités et par l'armée. Les progrès de la technique, diminuant considérablement le prix des ordinateurs, ont permis une diffusion de l'informatique dans les entreprises et dans les foyers.

L'avènement des réseaux de télécommunication a radicalement bouleversé le regard que nous avions sur l'informatique : les appareils étranges qu'étaient le téléphone et la télévision se sont vites intégrés dans nos sociétés. Les notions de temps et d'espace que nous connaissions grâce à la physique, ont-elles aussi été touchées par cette innovation.

Le monde des entreprises s'est vite rendu compte du potentiel des réseaux : délocalisation des succursales, centralisation de l'information, etc. Considérée au départ comme un avantage compétitif, l'informatique, à l'aube de l'an deux mille, est devenue un élément stratégique dans beaucoup d'entreprises.

L'année prochaine, en 1999, Internet fêtera son trentième anniversaire. Voilà déjà trente ans qu'Internet existe et permet à des réseaux du monde entier de s'interconnecter. Comportant uniquement quatre ordinateurs en 1969, Internet compte à l'heure actuelle un peu plus de vingt millions de sites.

Face à cette croissance plus exponentielle que linéaire, quelques questions viennent à notre esprit. Pouvons-nous rester maîtres de l'évolution continue et de la complexité croissante des réseaux ? Comment pouvons-nous garder le contrôle de l'interconnexion de milliers d'équipements hétérogènes ? Le domaine particulier de l'informatique qu'est « l'administration des réseaux » tente de trouver des solutions à ce problème.

Nous avons décidé de scinder ce mémoire en deux parties : la première permet d'aborder le côté théorique de l'administration tandis que la seconde se focalise sur le côté pratique et plus particulièrement sur l'étude de logiciels disponibles sur le marché.

La première partie de ce mémoire permettra d'aborder les concepts fondamentaux de l'administration des réseaux. Le premier chapitre introduira une propriété importante de l'administration : sa centralisation. Le deuxième chapitre

répondra à la question « qu'administre-t-on ? » ou, autrement dit, « que pouvons-nous administrer ». Les enjeux et les finalités de l'administration seront abordés dans le troisième chapitre, « pourquoi administrer ? ». Nous expliquerons la notion importante de qualité de service et nous montrerons que la conception actuelle des réseaux ne permet pas encore de rendre la négociation de la qualité de service totalement transparente. Le quatrième chapitre, « comment administrer ? », sera le plus technique de cette première partie. Le modèle informationnel et celui de communication des deux principaux organismes de normalisation, l'ISO (*International Standards Organization*) et l'IETF (*Internet Engineering Task Force*), seront décrits dans ce chapitre. Le modèle informationnel est nécessaire pour comprendre comment les composants du réseau sont perçus par l'administration. Le modèle de communication, supportant l'ensemble des flux de l'administration, est indispensable pour la centralisation des informations. Cette centralisation permet l'agrégation et la corrélation des informations de gestion nécessaires à la création de la vue globale présentée à l'administrateur du réseau.

Dans le sixième chapitre (le premier de la seconde partie), nous étudions les plates-formes d'administration « *ISM – Integrated System Management* » de la société Bull et « *HP OpenView* » d'Hewlett-Packard. Ces plates-formes ne sont pas des remèdes miracles applicables instantanément à tous les types de réseau, mais sont plutôt des boîtes à outils supportant des spécialisations et des développements qui répondent aux besoins spécifiques de chaque client. La distribution de logiciels, un exemple d'application développée sur la plate-forme ISM, sera exposée dans le septième chapitre. Le sujet de mon stage a porté sur la conception et la programmation d'une partie de cette application. La conclusion finale…

Remerciements

Au terme de ce mémoire, je tiens à témoigner ma gratitude à toutes les personnes qui, de près ou de loin, ont contribué à l'élaboration de ce mémoire.

Je remercie gracieusement M. Jean Ramaekers, mon promoteur, grâce à qui j'ai pu effectuer mon stage chez Bull et travailler sur un sujet qui m'intéressait beaucoup. Sa patience et ses conseils judicieux ont été pour moi un aide précieuse.

Je remercie également Christian Ritter pour m'avoir accueilli chez Bull et permis de vivre un stage enrichissant dont je garderai un excellent souvenir. Ses conseils, tant au niveau technique qu'au niveau humain, m'ont permis de surmonter les difficultés de mon travail en entreprise.

Je tiens aussi à remercier du fond du cœur tous les membres de l'équipe SDPC (Philippe, Pascale et Patricia) qui m'ont suivi tout au long de mon stage. Leurs

conseils, leur patience, leur générosité, la confiance qu'ils m'ont accordée et surtout leur bonne humeur quotidienne ont largement contribué à mon enrichissement personnel.

Merci à tous les membres de l'équipe ISM et en particulier à Alain, Christine, Didier, Dominique, François, Hoan, Josette, Marc, Olivier, Philippe B, Pierre A, Sophie et Thierry qui m'ont toujours prêté une attention particulière lorsque j'en avais besoin.

Je remercie cordialement Hugues Deghorain pour son accueil chaleureux et pour ses suggestions portant sur la réalisation de ce mémoire.

Merci également à tous les Enseignants et les Professeurs qui m'ont transmis leur savoir et leur expérience. L'accumulation de leur enseignement me permet, aujourd'hui, de prétendre au titre de licencié en informatique.

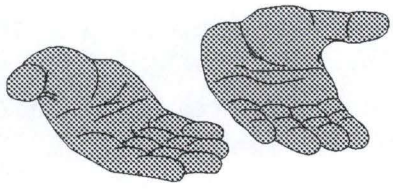
Conseils pour la lecture

Pour ne pas surcharger ce mémoire de définitions d'acronymes, nous avons préféré les regrouper dans une table que vous pourrez trouver à la page 88. Néanmoins, la traduction de certains d'entre eux peut aider à la compréhension ; ils seront soit définis en note de bas de page, soit dans le texte, entourés de parenthèses.

Nous avons également rassemblé l'ensemble des références bibliographiques à la page 91. Seules les citations extraites telles quelles d'un ouvrage sont référencées dans le texte avec la convention habituelle : [<nom de l'auteur> <date de parution>].

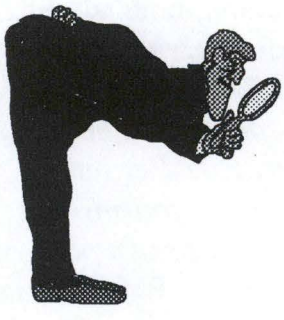
Le lecteur pourra consulter l'index situé à la fin de ce mémoire (page 96) afin de retrouver la définition ou les explications d'un terme ou d'un concept.

Nous vous souhaitons une bonne lecture...



Première Partie

LES CONCEPTS DE L'ADMINISTRATION



Les questions abordées sont :

- Qu'administre-t-on ?
- Pourquoi administrer ?
- Comment administrer ?

Chapitre 1 : Introduction de l'administration

Le besoin de réseaux de communication est né de l'intérêt de mettre à la disposition des utilisateurs, répartis géographiquement, des fonctions de traitement et des ressources informatiques. L'évolution substantielle des réseaux, tant au niveau des fonctionnalités qu'au niveau des vitesses de transfert, permet l'interconnexion de plus en plus d'équipements informatiques. Qu'il s'agisse de réseaux d'entreprise, de réseaux locaux ou de réseaux grandes distances supportant des applications distribuées, ceux-ci sont devenus tout à fait indispensables.

Si ces réseaux permettent la **distribution** des ressources, ce qui convient très bien aux organisations décentralisées, ils nécessitent une **centralisation** de leur contrôle. Il est difficile de concevoir qu'une personne soit responsable de chaque équipement du réseau. Cette personne devrait assurer le bon fonctionnement de "son" équipement. Heureusement, cette solution n'est pas envisageable pour deux raisons : premièrement, parce que les coûts engendrés seraient bien trop importants, et deuxièmement, parce qu'il n'est pas possible de gérer une ressource d'un réseau si on n'a pas une **vision globale** de celui-ci. Autrement dit, les réseaux créent deux courants antagonistes : un premier qui décentralise les ressources et un second qui centralise le contrôle et la gestion de celles-ci, comme l'illustre la Figure 1 ci-dessous.

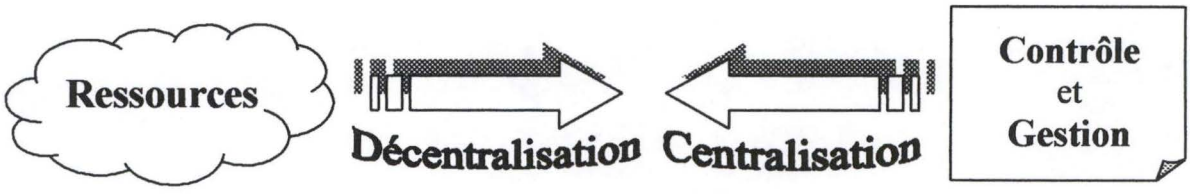


Figure 1 : Les deux courants antagonistes

Avant d'aborder les concepts fondamentaux, nous allons donner deux définitions de l'administration. Commençons par celle de [SIMONI & ZNATY 98] qui la définissent comme suit : « ... *c'est dire l'importance de la gestion, mais pas seulement dans son activité de surveillance des éléments du réseau, mais dans toutes les activités qui ont en charge le maintien du bon fonctionnement de l'ensemble des ressources mis en œuvre... L'administration du système doit alors fournir des mécanismes pour une exploitation interactive et dynamique, des automatismes pour contrôler, coordonner et planifier l'ensemble des ressources du système* ». Pour les auteurs, l'administration, synonyme de gestion, revient donc à considérer l'ensemble des tâches qui assure la qualité de service (QoS) du réseau.

La définition de [LABE] renforce cette notion de QoS : « *L'administration de réseaux privés a pour but la maîtrise des aspects techniques, financiers et organisationnels des réseaux ainsi que de la sécurité des accès à l'information. Vu du côté de l'utilisateur, l'administration de réseau aura pour principale conséquence de*

meilleures conditions d'utilisation du réseau, que ce soit au niveau de la continuité du service ou de la qualité de service (temps d'établissement, débits, délais de transit, qualité de transmission) ».

Maintenant que nous nous sommes fait une idée de ce qu'est l'administration, nous pouvons nous poser les questions fondamentales. Premièrement, nous devons identifier les composants qui seront pris en charge par l'administration en nous posant la question « ***qu'administre-t-on ?*** ». Ensuite, le « ***pourquoi administrer ?*** » permettra de dégager les enjeux et les finalités de l'administration. Finalement, nous pourrons étudier le « ***comment administrer ?*** », nécessaire à la compréhension des moyens mis en œuvre pour répondre aux diverses activités de l'administration.

Chapitre 2 : Qu'administre-t-on ?

L'administration a en charge l'ensemble des composants qui coopèrent pour rendre le service, c'est-à-dire tous ceux mis à la disposition de l'utilisateur final. *A priori*, tous les équipements réseaux ou connectés au réseau peuvent être administrés : les concentrateurs, les routeurs, les ponts, mais aussi les postes de travail, les machines Unix, etc.

Nous pouvons classer ces composants dans quatre catégories :

- les composants matériels (équipements),
- les composants matériels ou logiciels offrant la capacité de transmission (couche 1 à 3 du modèle OSI),
- les composants logiciels assurant le service (couche 4 à 7 du modèle OSI),
- les composants logiciels assurant les traitements applicatifs.

La Figure 2 représente un réseau simple. Nous donnons, pour chacun de ses éléments, des exemples de composants administrables.

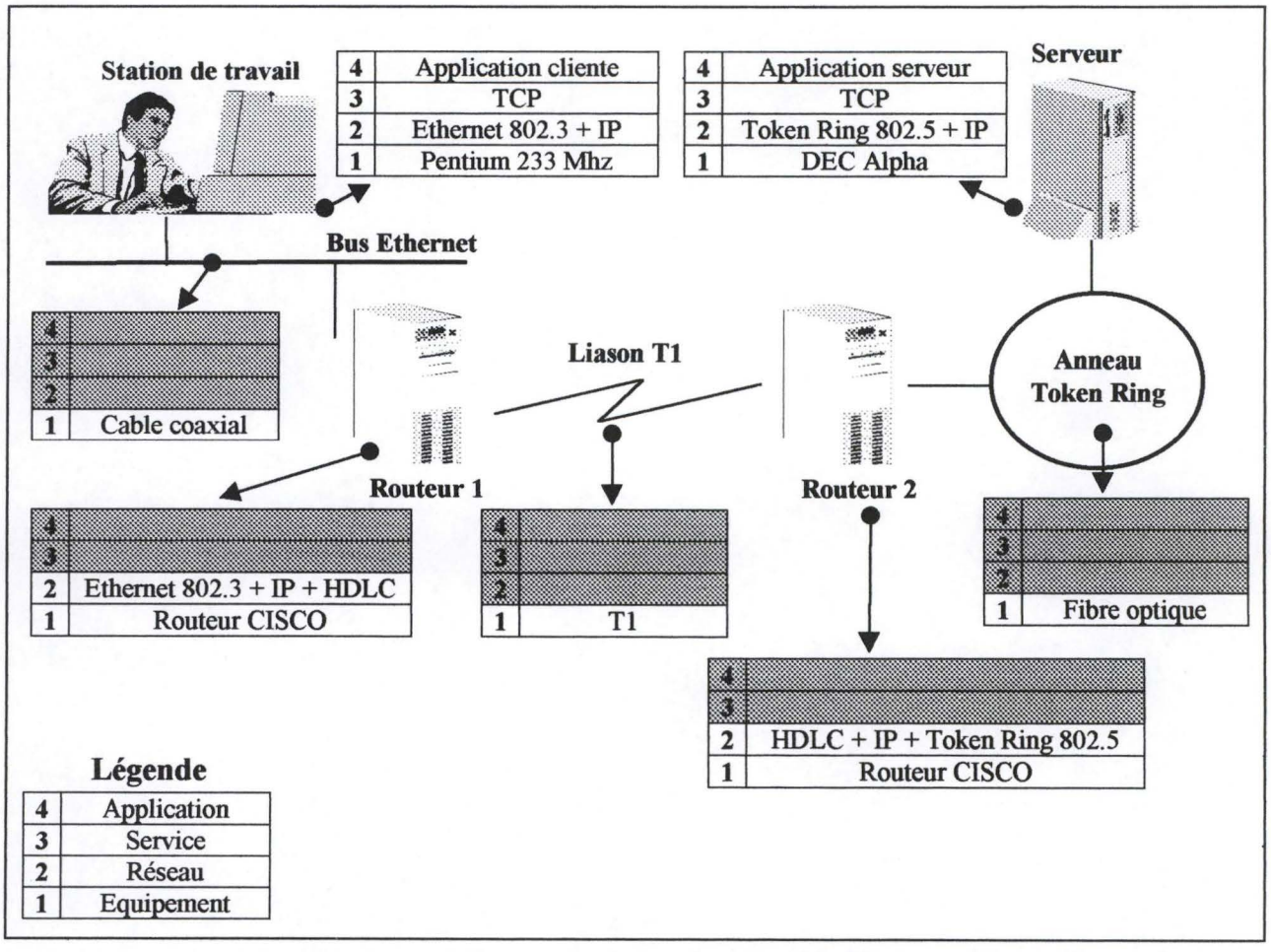


Figure 2 : Exemple de composants gérés par l'administration

Le Tableau 1 ci-dessous illustre, pour quelques éléments de la Figure 2, des exemples de questions que l'administrateur ou le client de l'application pourraient se poser.

Composant	Classe	Instance	Question
Station de travail	Application	Application cliente	Quel est le temps de réponse moyen d'une requête ?
	Service	TCP	Quel est le niveau de fiabilité de la connexion (taux d'erreur) ?
	Réseau	Ethernet 802.3	Quel est le taux de collision ?
		IP	Quelle est la table de routage ?
	Equipement	Pentium 233Mhz	Quels sont les processus en exécution ?
Liaison T1	Equipement	T1	Quel est le taux d'utilisation de la ligne ?

Tableau 1 : Exemple de questions techniques d'administration

Nous pouvons remarquer que les questions du Tableau 1 portent sur des domaines différents. Le temps de réponse et le taux d'utilisation sont des questions de performance tandis que l'acquisition d'une table de routage porte plutôt sur le domaine de la configuration.

Pour éviter que chaque constructeur crée ses propres domaines d'administration, l'ISO a créé une norme en proposant un modèle fonctionnel.

2.1 Le modèle fonctionnel de l'ISO

Afin d'exercer son activité, l'administration de réseaux s'appuie sur un certain nombre de fonctions qui peuvent être regroupées en cinq aires fonctionnelles ou SMFA (*Specific Management Functional Area*) dans la terminologie de l'ISO. Ce sont :

- la gestion des fautes,
- la gestion de la configuration,
- la gestion des performances,
- la gestion de la sécurité,
- la gestion de la comptabilité.

1. **La gestion des fautes** recouvre l'ensemble des fonctionnalités qui permettent de détecter, d'isoler et de corriger des pannes survenant sur un équipement. Les fautes proviennent de pannes de composants matériels ou logiciels et se manifestent par des événements particuliers (erreurs)

dans le fonctionnement du système. Lorsqu'une erreur est détectée, une analyse de l'état du système doit permettre de la localiser et de diagnostiquer sa cause. Une action curative permettant la reprise du fonctionnement du système doit suivre. Cette action peut être effectuée à distance (réinitialisation du système) ou, le plus fréquemment, par une intervention physique sur l'équipement. La gestion des fautes est une fonction d'administration vitale pour assurer aux utilisateurs un niveau de service satisfaisant du système.

2. **La gestion de la configuration** comprend les procédures permettant de modifier à distance la configuration de tous les équipements du réseau. L'objectif est de veiller au fonctionnement continu des services d'interconnexion. Il faut pouvoir démarrer, initialiser et arrêter le système ; positionner les paramètres du système ; recueillir des informations sur l'état du système et agir sur ces états ; associer des noms aux objets gérés. La configuration peut se faire ponctuellement sur demande de l'opérateur ou par téléchargement complet lorsque l'équipement a perdu sa configuration ou lors de sa réinitialisation.
3. **La gestion des performances** comprend les procédures de collecte des données et d'analyse statistique devant aboutir à la production de journaux de bord. Elle a pour but essentiel l'évaluation permanente du comportement du réseau afin de mesurer la qualité de service offerte et de prendre à tout moment les mesures nécessaires pour rester conforme aux objectifs. C'est-à-dire qu'elle doit s'accompagner de fonctionnalités permettant de mesurer et de comparer le niveau de performance du système, comme par exemple, des évaluations mathématiques du comportement des objets administrés et de l'efficacité de la communication.
4. **La gestion de la sécurité** couvre les procédures de contrôle d'accès (mots de passe), d'authentification (ce message a bien été envoyé par tel équipement), de cryptage ou encore de détection et d'historique des tentatives d'intrusion.
5. **La gestion de la comptabilité** a pour but de comptabiliser l'utilisation des ressources dans le but non seulement d'élaborer les factures liées à leur utilisation, mais aussi de connaître la répartition de cette utilisation (qui utilise quoi ?). Elle n'a, de façon instantanée, aucune influence sur le maintien de la qualité de service et l'obtention de bonnes performances.

[SIMONI & ZNATY 98] font remarquer que les activités de gestion regroupées en aires fonctionnelles font l'objet de débats, mais ne sont pas normalisées. Par contre, afin de faciliter leur mise en œuvre, l'ISO a défini une série de normes pour des fonctions de base, les SMF (*System Management Fonction*), dont le nombre ne cesse de croître.

Les cinq aires fonctionnelles utilisent les mêmes informations, les mêmes données avec simplement une précision ou une agrégation différentes. Par exemple une alarme (dépassement de seuil) sera analysée par la gestion des fautes pour être localisée et diagnostiquée ; elle sera comptabilisée par la gestion des performances pour les statistiques ; la gestion de la configuration en tiendra compte pour le routage et la reconfiguration des liaisons et des nœuds ; la gestion de la comptabilité pour éventuellement adapter la tarification ; et la gestion de la sécurité pour réagir dans le cas où la cause du dépassement serait frauduleuse.

2.2 Conclusion

L'administration consiste à surveiller et à maintenir le bon fonctionnement de tous les composants du réseau, à savoir les équipements, les éléments du réseau de transmission, les services et les applications. Mais aussi, elle contribue à ce que chaque réseau (ou sous-réseau) soit bien dimensionné et organisé.

Les cinq aires fonctionnelles définies par l'ISO permettent de "partitionner" le domaine d'activité de l'administration. Cette découpe conceptuelle est généralement utilisée pour la délégation des responsabilités entre les membres de l'administration et pour orienter la conception architecturale des outils de gestion. L'administration des réseaux ne peut se résumer à une de ces aires fonctionnelles ; en effet, elle serait incapable de remplir son principal objectif que nous allons définir dans le chapitre suivant.

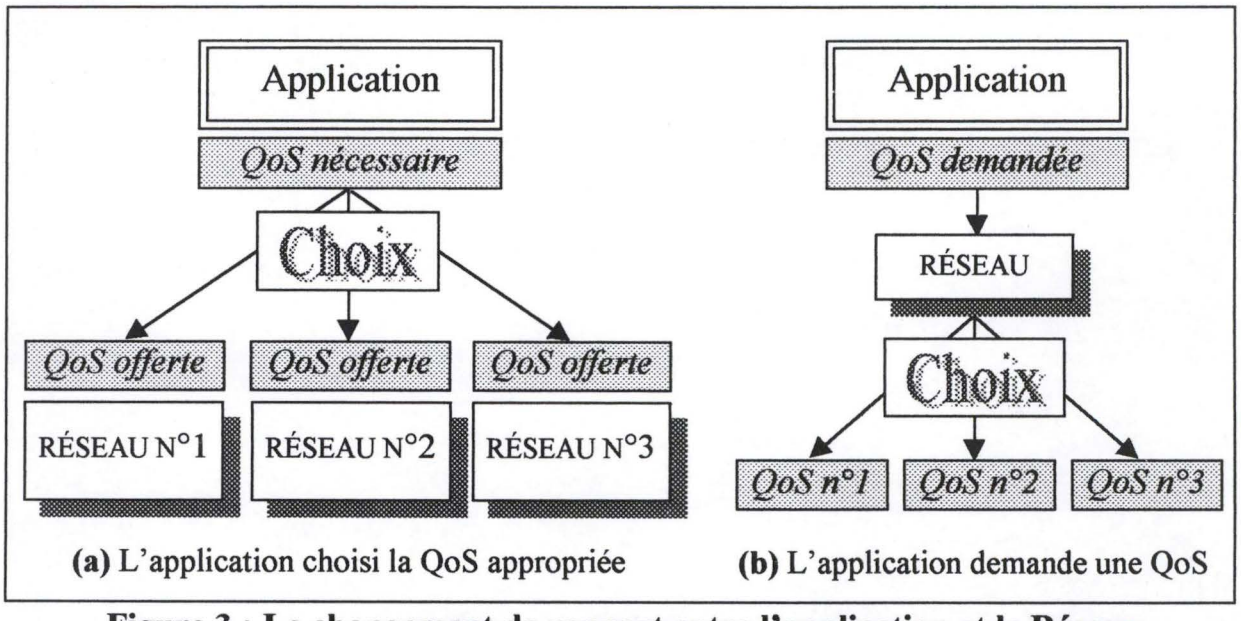
Chapitre 3 : Pourquoi administrer ?

Nous avons déjà abordé la notion de qualité de service dans l'introduction. La réponse à la question « pourquoi administrer » est donc simple : pour satisfaire une qualité de service demandée. En effet, le but final de l'administration se traduit par l'ensemble des activités permettant d'assurer la QoS du système, du réseau, conformément aux exigences des utilisateurs (l'administration n'est pas une fin en soi mais un moyen).

D'après [SIMONI & ZNATY,98], il est nécessaire, aujourd'hui plus que jamais, de prendre en compte la qualité de service demandée par l'utilisateur et exigée par chaque application. En effet, les flux sont transportés sur les mêmes supports mais toutes les applications n'ont pas les mêmes besoins. Jusqu'à présent, les opérateurs offraient une qualité de service qu'ils assuraient globalement et les utilisateurs choisissaient le réseau qui convenait *au mieux* à leurs besoins. Or, aujourd'hui, les technologies mises en œuvre commencent à pouvoir différencier les services offerts (pensons au RNIS-LB et à l'ATM). La relation *coûts-QoS* impose de satisfaire chaque qualité de service ni plus, ni moins.

3.1 Une nouvelle vision du Réseau

La Figure 3 ci-dessous illustre le changement du rapport entre le domaine des applications et le Réseau[1].



(a) L'application choisi la QoS appropriée **(b)** L'application demande une QoS

Figure 3 : Le changement de rapport entre l'application et le Réseau

[1] Le terme Réseau représente le support du transfert des informations et non un réseau particulier comme TCP/IP.

Dans le premier cas, Figure 3 (a), le concepteur de l'application détermine un niveau de QoS nécessaire, comme par exemple l'isochronisme[2], un délai de transfert maximum (10 milli-secondes) et une capacité de transmission minimale (64Kbps). Ensuite, le concepteur doit choisir le réseau le mieux approprié à ses critères de QoS. Ce choix est très difficile pour le concepteur car non seulement il doit tenir compte de nombreux paramètres (typologie du réseau, protocole, etc.) mais en plus, il est censé connaître l'ensemble des réseaux disponibles sur le marché.

Dans le deuxième cas, Figure 3 (b), le rôle du concepteur est totalement différent. Il établit les critères de QoS comme dans le premier cas et son travail s'arrête là. L'application conserve ses critères et c'est lors de son exécution qu'elle entame une phase de négociation avec le Réseau. Lors de cette phase, l'application soumet les critères de QoS requis au Réseau qui, par rapport à ses possibilités, détermine s'il est capable de répondre aux besoins de l'application. La phase de négociation peut aboutir à un compromis dans le cas où le Réseau ne peut assumer l'ensemble des besoins de l'application.

Le rapport entre l'application (ainsi que son concepteur) et le Réseau est fondamentalement différent. Dans le premier cas, l'application voit le Réseau comme un ensemble hétérogène de réseaux possédant chacun leurs propres qualités de service. Dans le deuxième cas, l'application voit le Réseau comme une seule entité atomique capable de négocier **les qualités fonctionnelles de son principal objectif** : transmettre des informations. Le choix du réseau possédant le niveau de QoS nécessaire est intrinsèque à l'application et devient transparent pour le concepteur. De plus, l'application n'est plus seulement conçue pour une classe de réseaux particulière mais est capable de s'exécuter sur tous ceux qui répondent aux besoins spécifiques de l'application. Le caractère dynamique de la négociation donne la possibilité au Réseau, dans le cas d'une surcharge, de rejeter la demande d'exécution de l'application.

3.2 La QoS des protocoles existants

Certains protocoles existants offrent déjà plusieurs niveaux de qualité de service. Le degré de raffinement dépend du protocole : cela va du plus élémentaire, comme TCP-UDP/IP, jusqu'aux plus raffinés, comme l'ATM, l'OSI ou le RNIS-LB.

Le Tableau 2 indique les quatre classes de service offertes par le RNIS-LB et des exemples d'utilisation. Ce tableau démontre que l'intégration de niveaux de QoS est bien une nouvelle préoccupation des concepteurs de protocoles. Malheureusement, le degré de raffinement reste faible : aucune notion de délai de transfert, de taux d'erreur, etc.

---

[2] **Isochrone** ou **isochronique** : adj. (grec *isos*, égal et *kronos*, temps). Qui s'effectue dans des intervalles de temps égaux. *Les oscillations isochrones du pendule*. Extrait du « Petit Larousse, édition 1997 ».

Service	Classe A	Classe B	Classe C	Classe D
Isochronisme	oui		non	
Débit	constant	variable		
Mode de connexion	mode connecté			non connecté
Exemple d'utilisation	voix à 64 Kbps	vidéo compressée	transfert de données	transfert de données sans connexion

Tableau 2 : Les classes de QoS du RNIS-LB

Il faut remarquer que, même si tous les protocoles proposent des niveaux de QoS avec un degré élevé de raffinement, il faut impérativement une intégration horizontale des protocoles pour que les bénéfices de la gestion de la QoS se ressentent au niveau applicatif[3]. Dans le cas contraire, on retombe sur le schéma (a) de la Figure 3. L'intégration horizontale signifie que l'application négocie le niveau de qualité en un seul point (le SAP) même si le Réseau comporte plusieurs piles protocolaires. C'est alors la couche interfacée avec l'application qui prend en charge la négociation, comme l'illustre la Figure 4 ci-dessous.

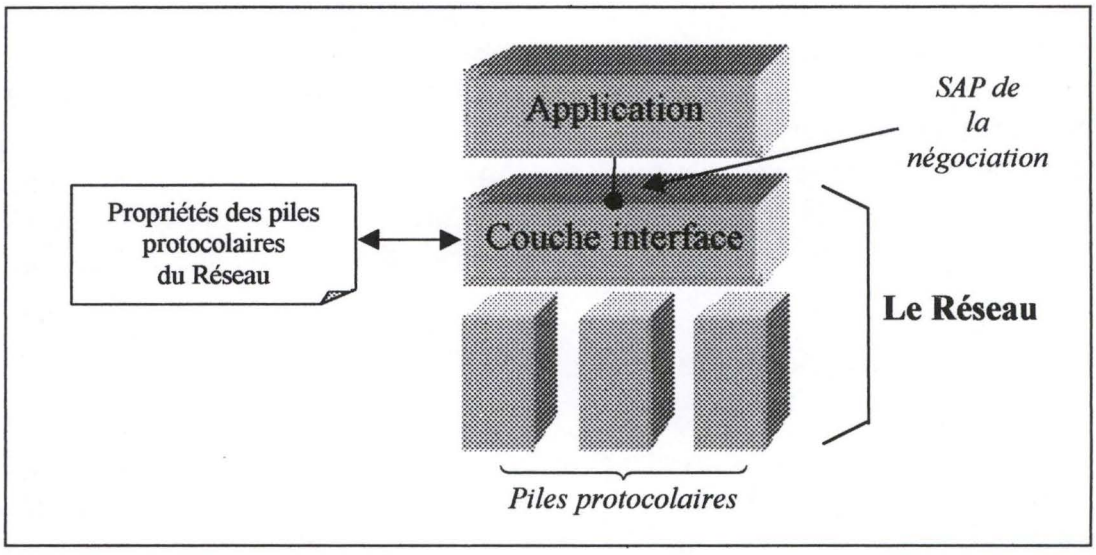


Figure 4 : L'interfaçage entre l'application et le réseau

Le rôle de la couche interface est de masquer à l'application la multiplicité des piles protocolaires. L'ISO a défini le modèle OSI en sept couches qui détermine, pour chacune des couches, leurs responsabilités fonctionnelles. Dans ce modèle OSI, la couche interface se situerait au-dessus de la ou les couches applications (couches 7). Cette couche d'intégration devrait, pour remplir sa tâche de négociateur, posséder les

[3] Le terme « applicatif » est synonyme d'« application ». Son utilisation dans le jargon de l'adminis-tration est cependant plus fréquent.

propriétés qualitatives et quantitatives des piles protocolaires sous-jacentes. C'est ici qu'intervient le rôle essentiel de l'administration. Nous reviendrons sur ce point dans la suite de ce chapitre. Analysons maintenant la définition de la QoS proposée par l'UIT-T (*Union Internationale des Télécommunications, secteur Télécommunications*) dans sa norme E-800.

3.3 La normalisation de la QoS

La recommandation E-800 de l'UIT-T fournit un cadre de description pour le concept de QoS. Elle précise entre autre deux définitions : celle de la *QoS attendue* par les utilisateurs et celle de la *qualité de fonctionnement* offerte par le réseau et qui représente les performances de ce dernier (*NP, Network Performance*).

La QoS attendue est définie comme suit :

Effet global produit par les caractéristiques d'un service fourni à un usager qui déterminent le degré de satisfaction que cet usager retire du service. La qualité d'un service est caractérisée par l'effet conjugué des notions suivantes : logistique de service, facilité d'utilisation du service, faisabilité, intégrité du service et d'autres facteurs propres à chaque service.

La qualité de fonctionnement (ou NP) est définie comme suit :

Aptitude d'un réseau ou d'un élément de réseau à assurer les fonctions liées à des communications entre usagers. Les performances du réseau contribuent à la faisabilité et à l'intégrité du service. La qualité de fonctionnement du réseau s'applique à la planification, au développement, à l'exploitation et à la maintenance assurée par le fournisseur du réseau. Les mesures de performances intéressent les prestataires de services et sont quantifiables sur la partie du réseau à laquelle elles s'appliquent. Elles expriment et représentent le comportement du réseau.

Dans la première définition, le terme « degré de satisfaction » induit que l'évaluation de la QoS est subjective et dépend fortement de l'utilisateur. Ces définitions contenues dans la norme E-800 abordent bien le sujet mais demandent une étude plus approfondie pour réussir une mise en œuvre des différents services et de leur gestion.

Afin de pouvoir apporter une appréciation la moins subjective possible de la qualité de service, il faut pouvoir l'exprimer à partir de grandeurs mesurables. Les mesures sont contrôlées et peuvent être recueillies, par exemple, à l'aide de sondes logicielles. Dans le paragraphe suivant, nous tenterons de définir des critères permettant de quantifier les performances d'un réseau.

3.4 Les critères de la QoS

Il n'y a pas de "bons réseaux" et de "mauvais réseaux" dans l'absolu. Un réseau est "bon" s'il répond aux services attendus. Un réseau basé sur des lignes à 9,6 Kbps, fonctionnant de 9h à 19h, peut être "bon" tandis qu'un réseau à 2Mpbs 24 heures sur 24 "mauvais". C'est pour cette raison que nous avons besoin de définir des critères.

La fonction principale d'un réseau est d'acheminer au travers d'un ensemble de nœuds et de liens des informations. Le transfert de ces informations se traduit par :

- la **capacité** à acheminer le volume déterminé par le *débit* des liens et la *capacité de traitement* des nœuds ;
- le **temps de transfert** maximum qu'il peut supporter, résultant du *temps de transit* et du *temps de traitement*, et qui correspond au temps global de réponse.

L'information ne peut être acheminée et traitée que si l'accès au service à travers le réseau est possible. Cette possibilité dépend de :

- la **disponibilité** du réseau : elle représente la période de temps pendant laquelle le service offert est opérationnel. A ne pas confondre avec l'accessibilité qui définit la façon dont la capacité du réseau est distribuée aux utilisateurs.

Une fois la connexion établie, la qualité du transfert dépendra d'une caractéristique très importante :

- la **fiabilité** du réseau : elle représente l'aptitude d'un système ou d'un équipement à fonctionner sans incident pendant un temps donné. C'est donc la probabilité pour un utilisateur de pouvoir mener à terme une session, sans interruption, avec un taux d'erreur acceptable et négocié pour son application.

3.5 La quantification des critères

Les critères que nous venons de définir ne s'appliquent pas uniquement à une entité protocolaire spécifique. Au contraire, ce sont des critères génériques qui se déclinent à chaque point d'accès d'un service (SAP). En effet, un réseau étant composé d'un ensemble de composants, la qualité de service offerte à une application dépend de l'"agrégation" des qualités de service de chacun des composants.

Illustrons notre propos par un exemple. Nous devons, dans un premier temps, modéliser un réseau. Nous ne retiendrons que les éléments existant sur la liaison entre un émetteur et un récepteur (liaison connue aussi sous le nom de *chemin*). Le modèle est représenté par la Figure 5.

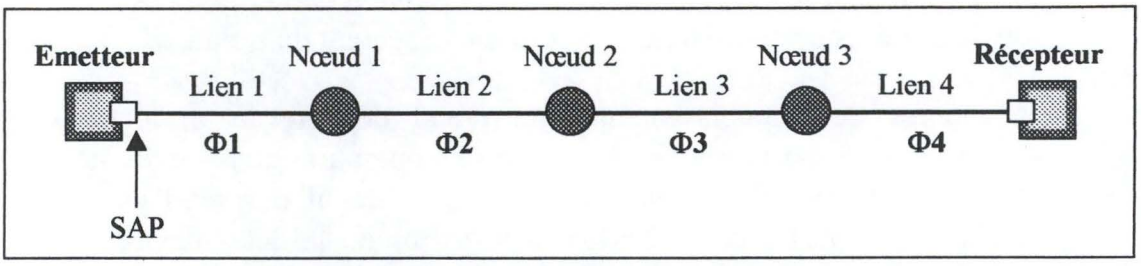


Figure 5 : Modèle d'un Réseau

L'intérêt de l'application est d'avoir des informations qualitatives et quantitatives sur le réseau à son point d'accès (SAP). Celles-ci peuvent se déduire des informations spécifiques de chacun des composants.

3.5.1 La capacité de transfert maximale

La **capacité de transfert** maximale offerte par le réseau est simplement la capacité du composant le plus lent. Ceci démontre que le composant le moins capacitaire crée le goulot d'étranglement du réseau.

> ➢ **Capacité maximale** = Minimum (C(composant)) ∀ composant

où un composant est soit un lien, soit un nœud ;
où C (lien) représente le débit de la ligne (la bande passante); et
où C (nœud) représente la capacité de traitement du nœud.

Le débit d'une ligne mesure le nombre d'octets utiles qui peuvent être transférés en une seconde. La capacité de traitement d'un nœud mesure le nombre d'octets qui peuvent traverser le nœud en une seconde. Le protocole utilisé, voire même la vitesse du bus de données du nœud, peuvent être déterminants dans le cas où les liens assurent un très haut débit.

3.5.2 Le temps de transfert

Le **temps de transfert** se calcule en sommant les temps de transfert de tous les composants formant le réseau. En effet, il représente le temps que met l'information émise par l'émetteur pour arriver jusqu'à son homologue, le récepteur. Cette grandeur dépend, évidemment, de la taille de l'information émise.

➢ **Temps de transfert** = $\sum$ T(composant) $\forall$ composant

où un composant est soit un lien, soit un nœud ;
où T(lien) représente le temps de franchissement de la ligne ; et
où T(nœud) représente le temps de franchissement du nœud.

Le temps de franchissement d'un nœud dépend du protocole utilisé. La sélection d'une ligne par un routeur n'est pas une opération gratuite. Remarquons aussi que le temps de transfert reste une approximation car il ne prend pas en compte le temps d'établissement de la connexion et le temps de la déconnexion. Si ces temps restent négligeables par rapport à la durée de la transmission alors ils ne doivent pas être pris en considération.

3.5.3 Evaluation de la capacité maximale et du temps de transfert

Nous pouvons évaluer les critères **capacité maximale** et **temps de transfert** sur le réseau modélisé à la Figure 5 au point d'accès offrant la capacité de transmission. Pour ce faire, nous devons attribuer arbitrairement des valeurs aux composants du réseau. Nous proposons les valeurs suivantes :

Lien	Débit	Temps de franchissement
Lien 1	10 Mbps	11 ns
Lien 2	100 Mbps[4]	4 ns
Lien 3	8 Mbps	10 ns
Lien 4	16 Mbps	13 ns
Nœud	**Capacité de traitement**	**Temps de franchissement**
Nœud 1	30 Mbps	1,5 ms
Nœud 2	40 Mbps	2 ms
Nœud 3	25 Mbps	2,3 ms

Tableau 3 : Données fictives pour les composants de la Figure 5

Capacité maximale = Minimum (C(composant)) $\forall$ composant
= Minimum (10,100,8,16,30,40,25) = **8 Mbps**

Autrement dit, l'application devra au maximum émettre des données à 8Mbps si elle désire avoir une continuité dans la transmission.

[4] N'oublions pas que le modèle du réseau ne reprend que les nœuds et les liens utilisés par l'émetteur et le récepteur. Le nœud 1 pourrait être un routeur qui multiplexe plusieurs liens en entrée sur le lien 2.

Le temps de transfert dépend de la taille de l'information émise et celle-ci peut être différente pour chaque application. Nous devons donc, pour notre estimation, fixer une taille arbitraire. L'extrapolation pourra ensuite être utilisée pour déterminer le temps de transfert d'autres volumes d'informations, à condition que ces volumes ne dépassent jamais septante pour cent de la capacité maximale du réseau. En effet, les valeurs extrapolées seraient faussées par l'éventuel écroulement (*trashing*) du réseau. Nous avons choisi de prendre la valeur de 1 Ko, soit 8192 bits, pour nos calculs.

$$\textbf{Temps de transfert} = \sum T(\text{composant}) \quad \forall \text{ composant}$$

$$= \sum T(\text{lien}) + \sum T(\text{nœud})$$

$$= [\ (1\text{Ko} / 10\text{Mbps} + 11\text{ ns}) + (1\text{Ko} / 100\text{Mbps} + 4\text{ns}) + (1\text{Ko} / 8\text{ Mbps} + 10\text{ns}) + (1\text{Ko} / 16\text{ Mbps}) + 13\text{ns})\ ] + [\ (1\text{Ko} / 30\text{Mbps} + 1{,}5\text{ ms}) + (1\text{Ko} / 40\text{Mbps} + 2\text{ms}) + (1\text{Ko} / 25\text{Mbps} + 2{,}3\text{ ms})\ ]$$

$$= \mathbf{8{,}89\ ms}$$

Autrement dit, il faudra entre 8 et 10 milli-secondes pour transporter 1 kilo-octets de l'émetteur jusqu'au récepteur.

Quelques remarques :

- Premièrement, le temps consommé par l'émetteur pour l'envoi des données et par le récepteur pour la réception ne sont pas considérés dans le calcul du temps de transfert. La question est de savoir si ces équipements font réellement partie du Réseau.
- Deuxièmement, le temps de transfert est basé sur l'hypothèse que toute la bande passante du Réseau est entièrement dédicacée à l'application. Pour remédier à cette hypothèse, il faut pouvoir collecter des données du Réseau en temps réel : c'est un des rôles essentiels de l'administration. Les valeurs refléteront alors l'état passé et présent du Réseau et seront extrapolées pour une vision de l'état futur de ce dernier. Il est évident qu'une extrapolation doit s'effectuer à partir d'un grand nombre de collections de données sinon les résultats obtenus risquent fortement d'être biaisés.
- Dernièrement, la transmission entre l'émetteur et le récepteur a été supposée sans erreur. La perte, la duplication ou la rémission de paquets ne sont pas impossibles. Toute chose étant égale par ailleurs, les erreurs du Réseau sont considérées par un autre critère, celui de la fiabilité.

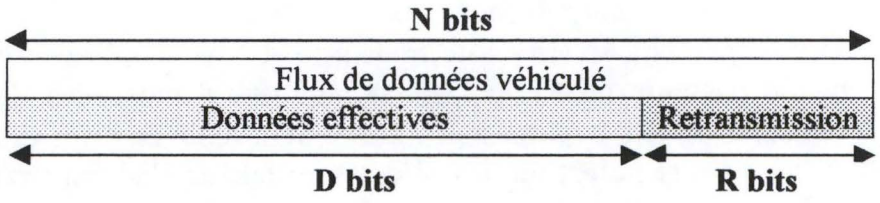
3.5.4 La fiabilité

La **fiabilité** d'un Réseau dépend essentiellement de deux choses : la fiabilité de ses composants et leur redondance. La fiabilité est donc une représentation du comportement du réseau ; elle implique la surveillance :

- des taux de perte, pendant la durée du transfert, au niveau de chaque composant de type nœud ;
- des taux de "déséquencement" et de duplication de l'information, suite au type de protocole mis en œuvre ;
- des taux d'erreur induits par la qualité des liens et traduisant des altérations de l'information.

La fiabilité dépend donc du taux d'erreur d'un lien et du taux de perte d'un nœud. La récupération des erreurs est généralement prise en charge par le protocole. La responsabilité du contrôle des erreurs est éclatée dans les différentes couches du modèle OSI. La couche transport (couche 4) a comme rôle de garantir une liaison fiable à sa couche supérieure. Sa complexité dépendra de la fiabilité offerte par le sous-réseau (couche 1 à 3).

Dans le cadre d'une connexion fiable, le flux de données véhiculé pour les besoins de l'application se compose de deux parties : une première qui regroupe les données de l'application et les valeurs ajoutées par les différentes couches du protocole (adresses, checksum...) et une deuxième partie qui correspond aux données retransmises suite à des erreurs.



Si le protocole utilisé par l'application gère les erreurs (par exemple TCP/IP), alors l'indicateur de la fiabilité de la transmission est le rapport entre le nombre de bits utilisés pour les retransmissions et le nombre total de bits transmis (rapport=R/D). Le rapport multiplié par cent donne le pourcentage de "déchets" de la transmission. Le rôle de l'administration est de collecter les données significatives pour le calcul du pourcentage moyen.

Si le protocole utilisé par l'application ne gère pas les erreurs (par exemple UDP/IP), alors l'indicateur de fiabilité est plus complexe et dépend du taux d'erreur, du taux de perte, etc.

3.5.5 La disponibilité

Le dernier critère, **la disponibilité**, est très important car il peut influencer considérablement la qualité du service offert à l'application. Un réseau peut être rapide et fiable mais indisponible ; autrement dit, l'application ne peut pas en profiter.

Sur un réseau de communication, le transfert de l'information peut être réalisé en mode « connecté » ou en mode « non connecté ». Le mode connecté consiste à vérifier et à réserver les ressources nécessaires au transfert de l'information (y compris l'appelé) avant d'envoyer les informations. Après la connexion, l'application dispose d'un canal de communication qui lui est réservé. Le mode non connecté permet quant à lui, d'envoyer directement des informations, sans vérifier au préalable que les ressources nécessaires sont disponibles.

Les échecs de connexion (en mode connecté) peuvent être comptabilisés dans un paramètre générique appelé *taux de rejet* ou encore *taux d'accessibilité*. Ces taux sont, pour l'application, des indicateurs sur la disponibilité du réseau. Par contre, en mode non connecté, le *taux de perte* dû à une saturation du réseau peut servir d'indicateur pour l'application. L'historique des différents taux prélevés peut servir de support aux différentes analyses portant sur le redimensionnement du réseau.

[BRON et al 91] font remarquer qu'une disponibilité du réseau de 95%, acceptée il y quelques années serait aujourd'hui refusée.

3.6 Conclusion

Nous avons vu que la finalité de l'administration d'un réseau était de garantir les qualités de service requises par les clients.

Les éléments d'appréciation du maintien du bon fonctionnement des réseaux de communication et des services se basent sur des critères génériques que nous avons décrit dans ce chapitre. Si la subjectivité renforce l'identité des êtres, elle n'est pas envisageable dans l'évaluation des réseaux. Par contre, les quatre critères « disponibilité, fiabilité, délai et capacité » permettent de quantifier et d'évaluer le fonctionnement d'un réseau et ceci d'une manière objective. De par leur caractère générique, les critères s'appliquent à tous les niveaux et offrent, par l'agrégation des différentes valeurs, une appréciation globale du réseau.

La programmation d'applications distribuées reste encore complexe à ce jour. L'hétérogénéité des réseaux en est la cause principale. Les organismes de normalisation, tel que l'ISO, ont créé des normes pour diminuer au maximum, voire annihiler, les problèmes causés par cette hétérogénéité. Malheureusement, ces normes ne portent que sur l'architecture des réseaux (modèle OSI en sept couches) et sur les interfaces d'accès. La diversité des qualités de services des différents protocoles reste

encore aujourd'hui un problème non résolu ; pourtant des solutions existent ! Nous avons proposé un modèle d'intégration basé sur une couche interface qui rend transparente la négociation des besoins de l'application en termes de qualité de service. L'élément important qui se dégage de ce modèle est que l'administration a un rôle important à jouer dans ce domaine.

Dans le monde des entreprises, une règle fondamentale est d'essayer de garantir aux clients la meilleure qualité possible des produits ou des services. Cette garantie tient compte du fait que les exigences des clients sont différentes. Alors, pourquoi le monde de l'informatique échapperait-il à cette règle ?

Chapitre 4 : Comment administrer ?

L'étude du « comment administrer ? », pour être complète, demande une étude détaillée qui risque de s'avérer très longue. Nous n'avons pas la prétention dans ce mémoire de décrire l'ensemble du fonctionnement de l'administration, de très bons livres référencés dans la bibliographie existent pour cela. Nous préférons décrire dans ce chapitre les grands principes de base et les deux modèles indispensables de l'administration. Commençons par les principes de base.

4.1 Les principes de base

Dans le cadre d'un réseau administré, chaque composant logique ou physique du réseau doit, pour pouvoir être administré, fournir une vue logique de son état. Autrement dit, il doit définir un ensemble de variables qui le représente. Cette vue logique est gérée par un processus appelé **agent**. Celui-ci joue un double rôle :

- premièrement, l'agent doit gérer la vue logique de telle façon que celle-ci représente le plus fidèlement l'état du composant (par exemple, en rafraîchissant périodiquement les données) ;
- deuxièmement, l'agent doit être capable de transmettre ses informations au **gestionnaire** (voir ci-dessous).

La vue logique permet donc de gérer l'hétérogénéité des composants. Par conséquent, l'administrateur perçoit les composants fonctionnellement identiques (les routeurs, les stations, les liens, etc.) d'une façon homogène. Il n'a pas besoin de connaître le constructeur et le modèle de l'équipement pour obtenir ses informations de gestion . Par exemple, tous les routeurs du réseau présentent la même vue logique contenant des valeurs spécifiques à chacun.

Nous avons vu, dans les chapitres précédents, que le support d'une bonne administration était une vision globale du réseau. Le rôle du gestionnaire est de créer cette vision globale en collectant les informations des composants par l'intermédiaire de leur agent, et de la présenter à l'administrateur du système, comme l'illustre la Figure 6 de la page suivante.

L'agent et le gestionnaire peuvent s'échanger des informations de deux façons : soit c'est le gestionnaire qui émet une **requête** vers l'agent, soit c'est l'agent qui émet une **alarme** vers le gestionnaire suite à un événement. Les requêtes sont utilisées par le gestionnaire pour collecter ou modifier des informations. Le concept d'alarme est semblable à celui des interruptions émises par les périphériques d'un ordinateur vers le processeur central. Par exemple, l'agent d'un routeur, détectant que le taux d'erreur a dépassé un certain seuil, peut émettre une alarme vers le gestionnaire.

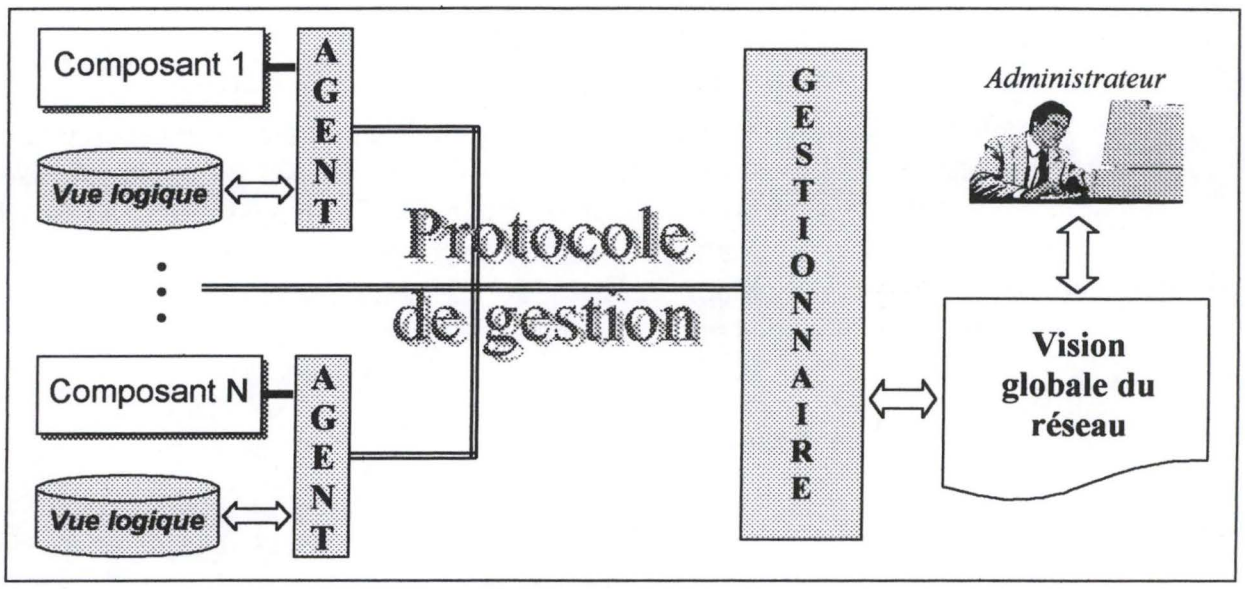


Figure 6 : Architecture de l'administration

La vue logique gérée par l'agent possède une structure particulière que nous détaillerons dans la section 4.2 intitulé « Le modèle informationnel ». L'échange d'informations entre les agents et le gestionnaire se fait par l'intermédiaire d'un protocole de gestion que nous approfondirons dans la section 4.3 intitulé « Le modèle de communication » .

Le rôle du gestionnaire est très complexe tant au niveau de la collecte des informations que de leur présentation à l'administrateur (généralement sous forme graphique). De plus, le gestionnaire doit être capable de modifier certaines informations d'un agent. Le gestionnaire a donc une structure beaucoup plus complexe que l'agent et se compose d'un ensemble d'entités responsables chacune d'une tâche particulière : c'est pourquoi il porte plutôt le nom **plate-forme d'administration**. Nous les étudierons en détail dans la seconde partie de ce mémoire.

L'ISO et la communauté Internet ont chacun développé leur propre technologie pour résoudre le problème de l'administration des réseaux. L'approche de la communauté Internet a été de créer des outils simples. Ceux-ci ont été conçus et déployés très rapidement ; aujourd'hui ils sont considérés comme le standard *de facto*. Par contre, l'ISO a préféré étudier le problème plus en profondeur. Leurs analyses ont débouché sur la création d'outils plus complets et plus génériques mais aussi plus complexes. A long terme c'est probablement leur solution, reconnue comme standard *de jure*, qui finira par s'imposer. Néanmoins, chacun des deux organismes propose un modèle informationnel et un modèle de communication que nous allons étudier dans ce chapitre.

4.2 Le modèle informationnel

Les structures de communication sont constituées de systèmes hétérogènes. En effet, la plupart des réseaux intègrent des produits et services provenant de différents constructeurs. Dans ce contexte, caractérisé par l'hétérogénéité des composants réseau et service, les acteurs principaux sont les organismes de normalisation qui ont fortement contribué à la définition d'un modèle informationnel.

- L'ISO fonde son modèle d'information sur l'approche orienté objet [ISO 10165-1][5]. De plus, cet organisme définit des informations semi-formelles pour la description des objets de gestion et leurs relations, et propose des librairies d'objets de gestion abstraits, donc réutilisables pour des architectures et des technologies spécifiques. Le modèle de l'ISO comporte deux composantes : le langage de description de classes d'objet de gestion et de relations [ISO 10165-5] [ISO 10165-7], et des librairies de classes réutilisables [ISO 10165-2] [ISO 10165-5].
- L'UIT-T, l'ETSI (*European Telecommunications Standard Institute*) et TINA-C (*Telecommunications Information Networking Architecture-Consortium*) s'appuient sur le modèle orienté objet défini par l'ISO, et définissent des modèles génériques pour la gestion des ressources d'un réseau de télécommunication. « Générique » car ces modèles entendent être applicables à différentes technologies de réseaux de télécommunication telles que l'ATM, le PDH, etc. Le modèle de l'UIT-T [UIT-T M.3100] comporte donc la composante librairie de classes réutilisables, le langage utilisé étant celui proposé par l'ISO.
- La communauté Internet modélise les données de gestion sous la forme d'attributs et de tables, dans un but de simplicité et de fourniture rapide d'une infrastructure opérationnelle de gestion de réseaux. Le modèle de l'Internet comporte deux composants : langage de description de données de gestion et MIB. La communauté Internet, à la différence de l'ISO et l'UIT-T, ne fonde pas son modèle sur l'objet. Le langage défini permet de directement produire des bases de données de gestion, les MIB, contenant des données spécifiques de la ressource à gérer.

Dans cette section, nous commencerons par étudier le modèle informationnel proposé par l'ISO pour ensuite étudier celui de la communauté Internet. Avant de conclure, nous proposerons une comparaison succincte des deux modèles.

[5] [ISO *n-x*] désigne une norme de l'ISO. L'ISO attribue un numéro unique à chaque norme (*n*) et à chaque document se référant à celle-ci (*x*). [ISO 10165-1] référence le premier document de la norme 10165.

4.2.1 Le modèle informationnel de l'ISO

Le modèle informationnel de l'ISO est orienté objet. Les concepts de ce modèle sont définis dans la norme relative au MIM (*Management Information Model*). Cette norme est décrite dans le document [ISO 10165-1]. Ce modèle est caractérisé par la définition des objets avec des propriétés spécifiques, ces objets étant les **objets de gestion** (*Managed Object*). Dans ce modèle, la vue logique gérée par l'agent (voir Figure 6 page 27) devient une collection d'objets de gestion.

Le fonctionnement interne de la ressource et les relations entre objets gérés et ressources ne sont pas connus de l'administration. Seules les caractéristiques la définissant en tant qu'objet administré sont accessibles par l'administration à travers son interface de gestion.

Les objets gérés ayant des propriétés identiques (structure et comportement) sont des instances de la même **classe d'objet**. Dans un contexte d'administration, il est nécessaire d'avoir une **relation de nommage** entre classes d'objet et un **mécanisme d'identification** de l'information. Nous étudierons ces concepts dans la suite de cette section.

L'ISO propose dans sa norme [ISO 10165-4] le GDMO (*Guidelines for the Defintion of Managed Objetcs*). On y trouve un ensemble de **formulaires** (*templates*) qui facilite la définition des informations d'administration.

4.2.1.1 Le langage de spécification

Le langage de spécification de l'ISO, appelé DMI (*Définition of Managed Information)* utilise le standard ASN.1[6] pour la description des objets. Les termes du langage DMI sont définis dans la norme [ISO 10165-2]. Nous décrivons les plus importants ci-dessous:

- **Objet de gestion** *(MOI - Managed Object Instance)* :
  C'est une représentation abstraite de la ressource qui peut être administrée par l'intermédiaire du protocole de gestion OSI.
- **Classe d'objet de gestion** *(MOC - Managed Object Class)* :
  C'est un ensemble nommé d'objets de gestion partageant le même ensemble d'attributs, de notifications et d'opérations.

[6] **ASN.1** (**A**bstract **S**yntax **N**otation **One**) est décrite par la norme ISO 8824. Les règles de codage des structures de données en une suite de données binaires sont décrites par la norme ISO 8825

- **Notification** :

  Les objets de gestion peuvent émettre des notifications (alarmes) à chaque occurrence d'événements détectés. Elles contiennent des informations relatives à ces événements survenus de façon soit interne, soit externe.

- **Attribut** *(Attribute)* :

  C'est une caractéristique de l'objet de gestion. Une valeur est associée à chaque attribut. On peut faire une déclaration sur la valeur d'un attribut particulier, cette **A**ssertion sur **V**aleur d'**A**ttribut (**AVA**) peut être soit vraie, soit fausse. A chaque attribut sont associés des droits d'accès GET, REPLACE ou GET-REPLACE signifiant que l'attribut est accessible en *lecture*, *écriture* ou *lecture-écriture*. Un attribut peut être multi-valué, c'est-à-dire, posséder un ensemble de valeurs. Dans ce cas, les deux droits d'accès ADD et REMOVE permettent d'ajouter ou de retirer une valeur à/de la liste de valeurs de l'attribut.

- **Groupe d'attributs** *(Attributes Group)* :

  Un groupe d'attributs est un moyen de référencer un ensemble d'attributs (généralement ayant la même sémantique).

- **Comportement** *(Behavior)* :

  La description du comportement d'un objet de gestion spécifie les caractéristiques dynamiques d'un objet et de ses attributs, les circonstances pour lesquelles les notifications doivent être émises et les actions. Elle inclut la sémantique des attributs.

- **Paquetage** *(Package)* :

  Un paquetage est une construction modulaire contenue dans un objet de gestion. Un paquetage possède une collection d'attributs, de notifications, d'opérations et de comportements. Le concept de paquetage a été introduit afin de faciliter l'étape de spécification des objets de gestion. La présence d'un paquetage peut être optionnelle ou obligatoire. Toutes les caractéristiques d'un paquetage obligatoire sont communes à toutes les instances d'une classe d'objet donnée. Les paquetages conditionnels sont présents si les conditions explicites qui leur sont associées sont satisfaites. Un paquetage peut être utilisé par plusieurs classes d'objet de gestion.

- **Héritage** *(Inheritance)* :

  C'est un mécanisme conceptuel par lequel une sous-classe acquiert les attributs, les notifications, les opérations et les comportements de sa super-classe.

Le GDMO définit les formulaires pour les types suivants : classe, paquetage, attribut, groupe d'attributs, notification, comportement, action, paramètre et lien de nommage. Les exemples ci-après illustrent l'utilisation de quelques-uns de ces formulaires.

Formulaire pour un attribut

<nom de l'attribut> **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX <type ASN.1> ← par exemple *integer*
MATCHES FOR EQUALITY ← opérations permises sur l'attribut.
BEHAVIOR <nom du comportement relatif à l'attribut>
REGISTERED AS {numéro d'enregistrement} ← identification unique de l'attribut[7]

Formulaire pour un comportement

<nom du comportement> **BEHAVIOR**
DEFINED AS <description> ← description de la sémantique de l'objet

Formulaire pour un groupe d'attributs

<nom du groupe> **ATTRIBUTE GROUP**
GROUP ELEMENTS <nom de l'attribut, ...> ← liste de noms d'attributs existants
DESCRIPTION <description> ← description de la sémantique du groupe
REGISTERED AS {numéro d'enregistrement}

Formulaire pour un paquetage

<nom du paquetage> **PACKAGE**
BEHAVIOR <nom du comportement relatif au paquetage>
ATTRIBUTES ↓ les attributs doivent avoir été définis auparavant
<nom de l'attribut> <statut ={lecture,écriture}> ;
état **GET** ; ← lecture seule autorisée
seuil **GET-REPLACE** ; ← lecture et écriture autorisées
...
ATTRIBUTE GROUPS
<nom du groupe, ...> ; ← les noms référencent des groupes existants
groupe-taille ;
groupe-taux ;
NOTIFICATIONS ↓ les noms référencent des notifications existantes
<nom de la notification, ...> ;
seuil_erreur_dépassé ;

REGISTERED AS {numéro d'enregistrement} ;

Formulaire pour une classe d'objet

<nom de la classe> **MANAGED OBJECT CLASS**
DERIVED FROM <super-classe>; ← super-classe dont la définition est héritée
CHARACTERIZED BY <nom paquetage> ; ← paquetage obligatoire
CONDITONNAL PACKAGES <nom paquetage,...> ; ← paquetage(s) optionnel(s)
(Les noms des paquetages référencés doivent avoir été définis auparavant)
REGISTRED AS {numéro d'enregistrement} ;

[7] L'identification des formulaires est expliqué à la page 35.

Formulaire pour une **notification**

<nom de la notification> **NOTIFICATION**
BEHAVIOR <nom du comportement relatif à la notification>
WITH INFORMATION SYNTAX <module information>
AND ATTRIBUTE IDS <nom de l'attribut, ...> ← liste des noms d'attributs existants qui seront transmis lors de l'événement

REGISTERED AS {numéro d'enregistrement}

4.2.1.2 La relation de nommage

Les différentes classes d'objet sont structurées selon la **relation de nommage** qui permet à une classe d'objet d'en contenir d'autres. La structure créée est **un arbre** dont chaque fils ne possède qu'un seul père (l'arbre est acyclique). Ce type d'arbre est familier aux informaticiens (arborescence des fichiers) et aux autres personnes (adresses postales).

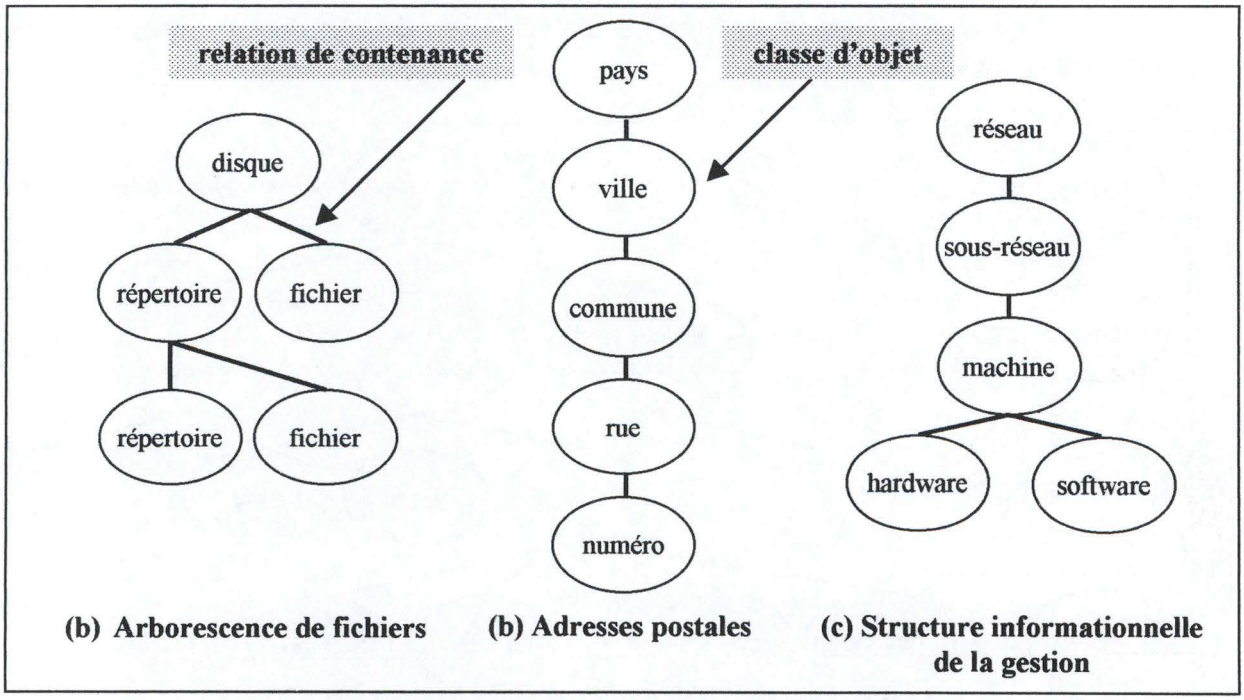


Figure 7 : Exemples de structure arborescente

Les différentes structures arborescentes de la Figure 7, constituées de classes d'objet reliées entre elles par la relation de contenance, signifient :

(a) qu'un disque contient des répertoires et des fichiers, et qu'un répertoire contient d'autres répertoires (récursivité) et des fichiers.

(b) qu'un pays contient des villes, qu'une ville contient des communes possédant un code postal, qu'une commune contient des rues et qu'une rue contient des numéros.

(c) qu'un réseau contient des sous-réseaux, qu'un sous-réseau contient des machines et qu'une machine contient du hardware et du software.

Les arbres de la Figure 7 sont appelés, dans le monde OSI, **arbres de nommage MIT** (*Management Information Tree*). Chaque classe d'objet se situe dans l'arbre par rapport à une classe d'objet supérieure dont elle est la subordonnée. Cette position est précisée au moment de la création de la classe d'objet par un **lien de nommage**. Un exemple est proposé à la fin de cette section.

Rappelons la différence entre une classe d'objet et un objet. Une classe d'objet est la description d'une structure d'information contenant des éléments (par exemple, des attributs, des méthodes, etc.) Une classe est donc décrite à l'aide d'un langage. Par contre, un objet est une portion de mémoire contenant des valeurs pour les éléments de la classe qu'il instancie. Un objet a donc une durée de vie dans un système, contrairement à la classe qui peut être considérée comme intemporelle.

La Figure 8 illustre un exemple d'instanciation de la structure informationnelle **(c)** de la Figure 7.

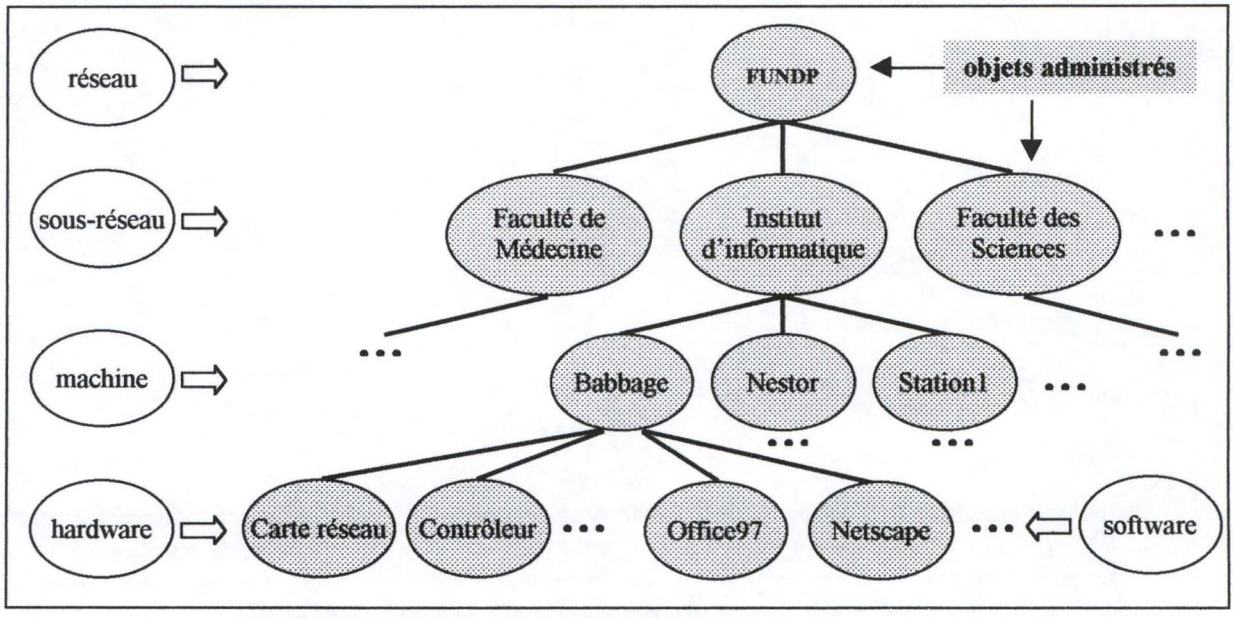


Figure 8 : Exemple d'instanciation d'une structure arborescente

Pour des raisons de non ambiguïté, on utilise, 1dans la terminologie de l'ISO, « relation de nommage » pour référencer la relation entre les classes et « **relation de contenance** » pour celle entre les instances (les objets de gestion). Les arbres portent également des noms distincts : « arbre de nommage MIT » pour celui des classes et « **arbre de contenance** » pour celui des instances.

Attention au fait que la relation de contenance ne correspond pas forcément à une relation de contenance physique entre les objets.

Le mécanisme d'identification pour un objet est basé sur une propriété importante : chaque classe d'objet doit posséder un attribut identifiant. Celui-ci permet d'identifier, sans ambiguïté, les objets appartenant à une même classe et qui sont rattachés à un même objet supérieur (autrement dit, à un même père).

La gestion OSI autorise deux formes de nommage, la forme locale et la globale. On appelle **RDN** (*Relative Distinguished Name*) d'un objet, la paire {attribut identifiant,valeur}. C'est donc le nom de l'attribut identifiant de la classe et sa valeur dans l'objet. Le RDN est le mécanisme de référence locale. Le **DN** (*Distinguished Name*) d'un objet permet d'identifier celui-ci au sein de l'arborescence. Il se compose de la séquence des RDN des objets situés sur le chemin allant de la racine jusqu'à l'objet référencé. Le DN est donc le mécanisme de référence globale d'un objet.

Par exemple, considérons que les classes d'objet de la Figure 8 (réseau, sous-réseau, machine...) possèdent toutes un attribut identifiant, appelé ID, et que les noms utilisés pour les objets représentent la valeur de cet identifiant. Dans ce cas, le RDN de la racine est {ID,"FUNDP"} et ceux du deuxième niveau {ID,"Babbage"}, {ID, "Nestor"} et {ID,"Station1"}. La Faculté des Sciences pourrait posséder une machine portant le nom Station1 sans pour autant enfreindre la propriété du RDN. En effet, les deux Stations1 seraient rattachées à un objet supérieur différent (l'un l'Institut d'informatique, l'autre la Faculté des Sciences). Pour référencer l'objet Netscape au sein de l'arborescence, nous devons utiliser son DN : {{ID,"FUNDP"}, {ID,"Institut d'informatique"}, {ID, "Babbage"}, {ID,"Netscape"}}.

Le formulaire *lien de nommage* (*name binding*) permet de spécifier la relation de contenance.

<nom du lien> **NAME BINDING**
SUBORDINATE OBJECT CLASS <classe subordonnée>;
NAMED BY SUPERIOR OBJECT CLASS <classe supérieure> ;
WITH ATTRIBUTE <attribut identifiant>; ◄——— Identifiant de la classe subordonnée
REGISTRED AS {numéro d'enregistrement} ;

4.2.1.3 L'identification des types d'information

Chaque formulaire (classe, attribut, etc.) décrit un type d'information particulier. Le nom du formulaire est utilisé pour désigner de manière unique un type d'information au sein du document dans lequel il est déclaré. Le nom est donc un mécanisme de référence locale exactement comme le nom d'une variable locale dans une fonction ou une procédure.

mécanisme de référence locale

```
<nom de l'attribut> ATTRIBUTE
WITH ATTRIBUTE SYNTAX ...
...
REGISTERED AS {numéro d'enregistrement}
```

mécanisme de référence universel

L'identifiant universel du type d'information (le numéro d'enregistrement) est attribué lors de l'enregistrement du formulaire. Il est unique au monde, il est véhiculé dans les protocoles de communication et il permet de reconnaître un type donné parmi tous ceux enregistrés. Les identifiants sont délivrés par des autorités d'enregistrement comme l'ISO ou l'UIT-T. Les identifiants, appelés aussi OBJECT IDENTIFIER, correspondent à une position dans un **arbre d'enregistrement** (*Registration Tree*) que nous décrirons dans la section suivante (page 37).

4.2.1.4 Conclusion

Nous avons vu qu'il existait quatre types d'arbres différents. Le premier est l'arbre de nommage MIT des classes d'objet, basé sur la relation de nommage. Cette relation a la propriété importante de contenir un attribut qui permet d'identifier les différentes instances de la classe subordonnée. Le mécanisme de référence d'une instance est basé sur le RDN et le DN. Les instances forment le deuxième arbre, l'arbre de contenance ; on parle alors de relation de contenance entre les objets. L'ensemble des types d'information est enregistré dans le troisième arbre, celui d'enregistrement. Chaque type d'information possède un identifiant unique et universel. Par exemple, l'administrateur qui désire recevoir les interfaces d'un routeur devra spécifier :

- premièrement, le numéro d'enregistrement de la classe d'objet qui modélise l'interface d'un routeur ;
- et deuxièmement, le DN du routeur qui sera certainement constitué de l'adresse réseau du routeur.

La relation d'héritage entre les classes d'objet forme le quatrième et dernier arbre : l'arbre d'héritage.

4.2.2 Le modèle informationnel de la communauté Internet

Pour le modèle informationnel de la communauté Internet, l'IETF (*Internet Engineering Task Force*) a défini deux RFC (*Request For Comment*). Il s'agit du [RFC-1155] qui décrit la structure et le nommage de l'information de gestion, à savoir SMI (*Structure of Management Information*), avec une extension définie dans le [RFC-1212]. Quant au [RFC-1213], il décrit la base d'information MIB-II. Ce dernier RFC enrichit le [RFC-1158] qui décrit une première version de la MIB, la MIB-I. L'ensemble des RFC est disponible sur le WEB[8]. Le modèle informationnel défini par la communauté Internet est simple et ceci dans le but de fournir rapidement une infrastructure opérationnelle de gestion de réseaux. Bien sûr, cette simplicité entraîne des faiblesses conceptuelles du modèle. Nous rappelons que ce modèle n'est pas orienté objet. Nous commençons notre étude par le langage de spécification SMI.

4.2.2.1 Le langage de spécification

Le langage de spécification du monde Internet se base uniquement sur un sous-ensemble du langage ASN.1 pour des raisons de simplicité.

Chaque type d'objet (correspondant à la classe d'objet dans le modèle de l'ISO) possède un nom, une syntaxe et un encodage.

- **Le nom** d'un type d'objet est représenté de manière unique par l'identificateur d'objet (*OBJECT IDENTIFIER*).
- **La syntaxe** d'un type d'objet définit la structure abstraite de données qui modélise la ressource.
- **L'encodage** d'un type d'objet est une règle qui explique comment les instances de ce type d'objet sont encodées.

Les noms sont utilisés pour identifier les objets gérés. Chaque objet possède un identificateur d'objet qui lui est propre. Les identificateurs ont une structure hiérarchique, arborescente. L'identificateur d'objet est une séquence de nombres entiers qui parcourent un arbre (voir Figure 9). L'IETF utilise l'arbre d'enregistrement défini par l'ISO et l'UIT-T afin de nommer l'information de gestion. Cet arbre est composé d'une racine à laquelle sont liés des nœuds marqués. Un nœud est identifié de manière unique, son nom correspond à la concaténation des nombres (ou des noms) qui relient la racine au nœud considéré (à lui-même). Chaque nœud peut avoir des fils qui sont aussi marqués. La marque de chacun des nœuds se compose de la brève description textuelle et d'un nombre entier. Seul le nœud racine de l'arbre n'est pas marqué. Son rôle est uniquement de créer un seul arbre à partir d'une forêt (les trois

[8] Le lecteur pourra trouver, à la page 94, les adresses des sites WEB où sont disponibles les RFC.

sous-arbres qu'elle contient) et ceci parce que le mécanisme d'identification impose l'unicité de la racine.

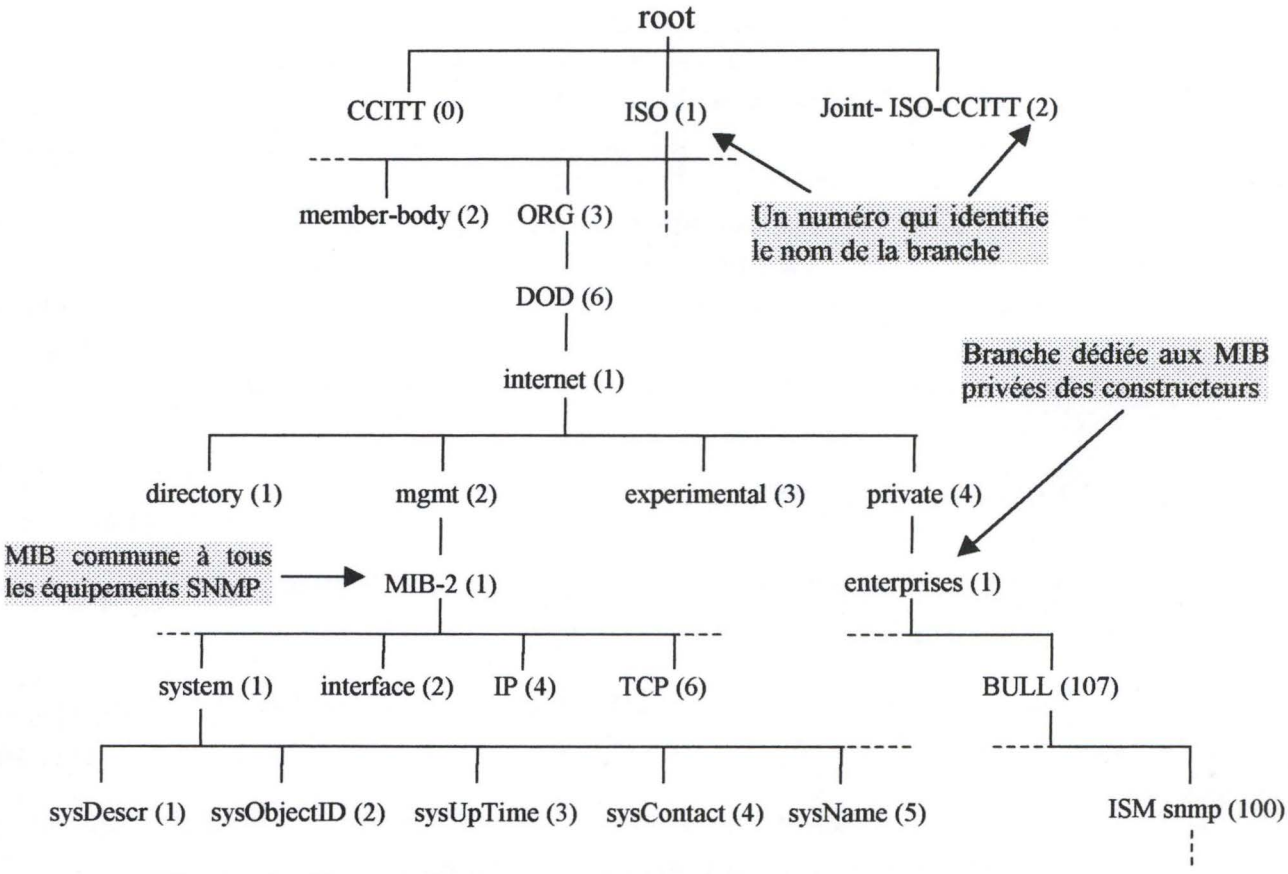


Figure 9 : Extrait de l'arbre d'enregistrement de l'ISO / UIT-T

Les nœuds de l'arbre structurent l'information tandis que les feuilles représentent les informations (on utilise parfois les termes variable et tableau).

Le nœud racine contient trois fils. Les deux premiers sont gérés respectivement par l'UIT-T, autrefois appelé CCITT, et l'ISO, tandis que le troisième est géré par les deux organismes.

L'ISO a créé et dédié un sous-arbre, ORG(3), aux organismes de normalisation nationaux et internationaux. Deux des sous-arbres de ORG(3) ont été assignés au NIST (*U.S. National Institutes of Standards and Technology*). La gestion d'un de ces sous-arbres, DOD(6), a été transférée au ministère de la défense DoD (*Departement Of Defense*). Le DoD a alloué un nœud à la communauté Internet, internet(1), et c'est l'IAB[9] qui a été nommé responsable de ce sous-arbre.

[9] L'IAB (*Internet Activities Board*) est le groupe technique chargé de surveiller le développement des protocoles de l'Internet. Il est divisé en deux entités : l'IETF (*Internet Engineering Task Force*), qui est responsable de la conception à court terme et de la création des standards et l'IRTF (*Internet Research Task Force*), qui est responsable de la recherche à long terme.

L'ensemble des objets d'administration de la communauté Internet sera donc contenu dans le sous-arbre identifié par le nœud ISO(1).ORG(3).DOD(6).internet(1). L'IAB a créé, à partir de ce nœud, quatre branches : directory(1), mgmt(2), experimental(3) et private(4).

- Le sous-arbre directory(1) est réservé pour l'utilisation de l'OSI Directory dans le monde Internet. Il n'a pas encore été implémenté car les discussions sont toujours en cours.
- Le sous-arbre mgmt(2) est consacré à la gestion des réseaux. L'ensemble des objets standards d'administration définis par le monde Internet, autrement dit la MIB standardisée, est contenu dans ce sous-arbre.
- Le sous-arbre experimental(3) est utilisé pour l'expérimentation. L'IAB a délégué sa responsabilité à différentes autorités. Certaines extensions non encore normalisées de la MIB se retrouvent dans ce sous-arbre.
- La branche *private(4)-enterprise(1)* est très développée car elle contient des MIB propres à chaque constructeur et à chaque équipement au sein de sa gamme de matériels. Il existe une branche par constructeur : par exemple la société BULL porte le numéro 107 à partir de l'arborescence *1.3.6.1.4.1.*

La branche MIB-2(1), rattachée au nœud mgmt(2), définit les informations minimales communes à tous les agents SNMP[10]. Elle contient des objets simples tels que le nom de l'équipement, son adresse IP, etc. Avant d'aborder des exemples d'objets, analysons la syntaxe du langage SMI.

La syntaxe est utilisée pour définir la structure de données qui correspond au type d'objet. Un ensemble limité de constructions ASN.1 est utilisé afin de définir cette structure.

Les informations que représentent les objets, peuvent être représentées à travers trois **types simples** : INTEGER, OCTECT STRING et OBJECT IDENTIFIER. Ces derniers sont aussi appelés dans la littérature « types non agrégés ».

Le constructeur SEQUENCE du langage ASN.1 est le seul **type agrégé** autorisé du langage SMI. Il permet de construire soit des listes, soit des tableaux. Pour les listes, la syntaxe a la forme *SEQUENCE { <type1>, ..., <type N> }* où chaque *<type i>* correspond à un des types simples. Pour les tables, la syntaxe prend la forme *SEQUENCE OF <entrée>*, où *<entrée>* désigne un constructeur de liste.

Le langage SMI propose aussi un ensemble de types prédéfinis basés sur les types simples : *NetworkAddress* pour représenter des adresses de protocoles, *IpAddress* pour représenter des adresses Internet, *Counter* pour définir des compteurs

[10] **SNMP**, signifiant **S**imple **N**etwork **M**anagement **P**rocotol, est le protocole de gestion du monde Internet que nous décrirons dans le paragraphe 4.3.2 intitulé « Le modèle de communication de la communauté Internet ».

de type entier, Gauge pour caractériser un entier positif ne dépassant pas une taille maximale, *TimeTicks* pour compter le temps en centième de seconde, et finalement *Opaque* qui représente une syntaxe quelconque encodée sous la forme d'un OCTET STRING.

La description d'un **type d'objet** (l'équivalent d'une classe d'objet dans le modèle de l'ISO) se réalise au moyen d'un formulaire défini par le langage SMI. Ce formulaire se compose de cinq entrées :

- **OBJECT** : un nom textuel (nommé descripteur d'objet) pour le type d'objet avec un identificateur d'objet (OBJECT IDENTIFIER) correspondant.
- **SYNTAX** : la syntaxe de l'objet est décrite avec le langage ASN.1
- **DEFINITION** : la définition de l'objet consiste en une description textuelle de la sémantique de l'objet.
- **ACCESS** : le type d'accès autorisé sur l'objet. Les valeurs possibles sont *read-only*, *read-write*, *write-only* ou *not-accessible.*
- **STATUS** : *mandatory*, *optional* ou *obsolete*.

Pour illustrer cette définition, voici deux exemples d'utilisation du formulaire SMI :

```
        ↓                    descripteur du type d'objet
varEntière OBJECT-TYPE
  SYNTAX integer
  DEFINITION "Ceci est la définition d'une variable entière"
  ACCESS read-write
  STATUS mandatory
::= { père 1 }  ←————   identificateur du type d'objet (OBJECT IDENTIFIER)
```

```
tableau OBJECT-TYPE
  SYNTAX SEQUENCE OF entréeTable
  DEFINITION "Ceci est la définition d'une table"
  ACCESS read-only
  STATUS mandatory
::= { père 2 }

entréeTable OBJECT-TYPE
  SYNTAX EntréeTable
  DEFINITION "Ceci est la définition d'une entrée de la table"
  ACCESS read-only
  STATUS mandatory
  INDEX champ1  ←———— l'index spécifie le champ de la table qui permet d'identifier
::= { tableau 1 }          les tuples de celle-ci.

EntréeTable ::= SEQUENCE {
     champ1    integer,
     champ2    OCTECT STRING,
     champ3    ipAddress   }
```

4.2.2.2 L'identification de l'instance d'un type d'objet

Il faut distinguer la notion de *type d'objet* et d'*instance d'objet* (parfois appelé *objet* par abus de langage). Le type d'objet, l'équivalent de la classe d'objet dans le modèle ISO, est la description d'une structure de données.

Par contre, l'instance d'objet est un cas particulier du type qu'il instancie ; il possède une valeur. Chaque instance est identifiée par deux éléments : identificateur du type de l'objet et identificateur de l'instance.

On peut identifier un type d'objet de trois façons différentes. Par exemple, le type d'objet *sysDescr* (voir Figure 9, page 37) peut être identifié par son identificateur (la séquence 1.3.6.1.2.1.1.1), par son descripteur (sysDescr) et aussi par la séquence formée du descripteur du nœud père et de l'entier l'identifiant parmi les fils de nœud (system 1).

L'identification de l'instance est différente selon que l'instance est une variable ou un tableau. Pour une variable, on suffixe l'identificateur du type de l'objet par ".0". Par exemple, la variable *sysDescr* est identifiée par la séquence 1.3.6.1.2.1.1.1.**0**. Par contre, pour un tableau, le suffixe est la valeur de l'index du tuple auquel on désire accéder.

4.2.2.3 La Base d'Information de Gestion (MIB)

La communauté Internet a défini une collection d'objets « standard » à administrer à travers la spécification des MIB-I et MIB-II. Les objets définis ont une structure particulièrement simple. En effet, la définition des objets est limitée à un type simple ou à une table d'objets de type simple.

La MIB-I correspond au premier lot de définitions d'objets SNMP. Elle contient une centaine d'objets, rangés par groupes fonctionnels au nombre de huit : *System*, *Interfaces*, *Address Translation (AT)*, *Internet Protocol (IP)*, *Internet Control Message Protocol (ICMP)*, *Transmission Control Protocol (TCP)*, *Unreliable Datagram Protocol (UDP)*, *Exterior Gateway Protocol (EGP)*. Ces huit groupes permettent uniquement de gérer un réseau TCP/IP. Certains de ces groupes sont obligatoires (les cinq premiers) ; d'autres sont considérés comme obligatoires si l'équipement à gérer fonctionne avec un protocole particulier. Par exemple, si un routeur utilise EGP, alors le groupe EGP est obligatoire.

Afin de descendre plus finement dans les fonctionnalités d'un équipement, de nouveaux objets ont été ajoutés progressivement depuis deux ans, définissant ainsi la MIB-II. Par exemple, le groupe *System* se voit doté de nouveaux objets (entre autre, *sysContact* qui spécifie le nom de la personne physique à contacter en cas de défaillance de l'équipement). De plus, de nouveaux groupes ont été ajoutés, tel SNMP

qui possède une trentaine d'objets de type simple utilisés pour gérer les performances du protocole de gestion.

Dans le monde de l'administration, le mot MIB est utilisé pour désigner deux concepts différents. Il désigne la description de la structure de données (décrite avec le langage SMI) et la collection de données qui se trouve sur l'équipement et qui est gérée par un agent. Nous préférons, pour éviter toute ambiguïté, utiliser le terme « schéma de la MIB » pour la description et « MIB » pour la collection de données (n'oublions pas que le B de MIB signifie Base).

Nous présentons ci-dessous un bref extrait du [RFC-1213] qui contient la description de la MIB-II.

```
RFC1213-MIB DEFINITIONS ::= BEGIN
 -- Extracted from RFC 1213
IMPORTS
        mgmt, NetworkAddress, IpAddress, Counter, Gauge,TimeTicks
        FROM RFC1155-SMI
        FROM RFC-1212 ;
...
mib-2 OBJECT IDENTIFIER ::= { mgmt 1}

-- groups in MIB-II
system      OBJECT IDENTIFIER   ::= { mib-2 1 }
interfaces  OBJECT IDENTIFIER   ::= { mib-2 2 }
at          OBJECT IDENTIFIER   ::= { mib-2 3 }
ip          OBJECT IDENTIFIER   ::= { mib-2 4 }
icmp        OBJECT IDENTIFIER   ::= { mib-2 5 }
tcp         OBJECT IDENTIFIER   ::= { mib-2 6 }
upd         OBJECT IDENTIFIER   ::= { mib-2 7 }
egp         OBJECT IDENTIFIER   ::= { mib-2 8 }

-- the System group.
-- Implementation of the System group is mandatory for all systems. If an
-- agent is not configured to have a value for any of these variables, a string
-- of length 0 is returned.

sysDescr OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
```

Début de la définition de la MIB-II

Importe des définitions d'un autre fichier

Définitions des groupes de la MIB-II

Identification de chaque groupe au sein de la branche MIB-II

Ensuite vient la définition de tous les objets de la MIB-II

```
                "A textual description of the entity. This value should include the full
                name and version identification of the system's hardware type,
                software operating-system, and networking software. It is mandatory
                that this only contain printable ASCII characters."
        ::= { system 1}
...

sysContact OBJECT-TYPE
        SYNTAX DisplayString (SIZE (0..255))
        ACCESS read-write
        STATUS mandatory
        DESCRIPTION
                "The textual identification of the contact person for this managed
                node, together with information on how to contact this person. "
        ::= { system 4}
.......
END◄————————————  Fin de la définition de la MIB-II
```

Le monde Internet a également spécifié, dans le document [RFC-1214], la MIB-II dans le langage DMI de l'ISO. Les conventions utilisées sont les suivantes :

- les groupes de la MIB-II deviennent des classes d'objet,
- les tables de la MIB-II deviennent aussi des classes d'objet,
- tous les autres objets deviennent des attributs.

Par exemple, la description suivante de la table :

```
tcpConnEntry  OBJECT-TYPE
    SYNTAX TcpConnEntry
    ...
TcpConnEntry ::=
    SEQUENCE
        {
        tcpConnState INTEGER,
        tcpConnLocalAddress ipaddress,
        tcpConnLocalPort INTEGER (0..65535),
        tcpConnRemAddress ipaddress,
        tcpConnRemPort INTEGER (0..65535)
        }
```

devient dans le langage DMI de l'ISO :

```
tcpConnEntry MANAGED OBJECT CLASS
    DERIVED FROM top ;
```

```
CHARACTERIZED BY
 tcpConnEntryPkg PACKAGE
 ATTRIBUTES
       tcpConnId              GET,
       tcpConnState           GET-REPLACE,
       tcpConnLocalAddress    GET,
       tcpConnLocalPort       GET,
       tcpConnRemAddress      GET,
       tcpConnRemPort         GET ;;;
REGISTERED AS ...
```

Certains attributs, comme *tcpConnId*, ont dû être ajoutés pour des raisons de conformité aux conventions du protocole de gestion de l'ISO.

En conclusion, une MIB est une collection d'objets de gestion, de la même façon qu'une base de données relationnelle est une collection de tables. La description des objets (appelée également type d'objet) s'effectue à l'aide du langage SMI. Pour créer une MIB, il est nécessaire de compiler sa description contenue dans un fichier texte. Le résultat de la compilation est un fichier de format binaire (la MIB) plus compact et directement interprétable par les outils d'administration.

4.2.3 Modèle de l'ISO vs Modèle de la communauté Internet

Le modèle informationnel proposé par l'ISO se fonde sur l'approche orientée objet tandis que les objets définis par l'IETF dans le monde Internet sont décrits sous la forme de variables simples mises dans un arbre d'enregistrement. Les avantages et les inconvénients de chacune des solutions sont les suivants :

- le modèle informationnel de l'ISO est plus complexe et difficile à appliquer. Des outils appropriés doivent être utilisés afin de créer des librairies d'objets de gestion OSI ;
- les propriétés d'héritage et d'abstraction associées à l'orienté objet, peuvent être utilisées afin de créer une structure d'objet bien définie ;
- les nombreuses librairies d'objets associées aux MIB créées par l'IETF montrent clairement qu'elles ne fournissent pas de structuration suffisante de l'information. En particulier, l'absence de mécanisme d'héritage pour la réutilisation signifie que des informations dérivées les unes des autres peuvent être contenues dans des sous-arbres différents de l'arbre d'enregistrement.

Le choix est donc simple : soit un modèle informationnel complexe basé sur des concepts riches de structuration, celui de l'ISO, soit un modèle informationnel facile d'utilisation mais incomplet, celui de la communauté Internet.

4.2.4 Conclusion

Nous avons abordé la dimension informationnelle en premier lieu car elle matérialise la connaissance du réseau et des services. Cette connaissance est essentielle pour la gestion et elle concerne aussi bien les aspects statiques que les aspects comportementaux. La puissance du modèle informationnel repose sur la représentation objet des composants réels (la vue logique). L'ISO fournit essentiellement un guide, les formulaires GDMO, tandis que la communauté Internet nous propose des bases d'informations pour les protocoles de communication.

Maintenant que nous savons représenter les réseaux et les services, il nous faut étudier comment constituer la base de connaissance centrale, autrement dit, la vue globale de l'administrateur. Cette étude porte sur le modèle de communication.

4.3 Le modèle de communication

Nous avons distingué dans l'architecture de l'administration (Figure 6, page 27), deux entités distinctes : l'agent qui possède l'information du composant, et le gestionnaire qui présente de manière conviviale cette information à l'administrateur. Il y donc un échange d'informations entre les deux entités. Le modèle de communication permet de décrire comment ces échanges sont réalisés. Il comprend la description de l'interface d'accès aux informations, constituée de différentes requêtes, et le protocole de gestion qui s'occupe du transfert des données.

Le monde ISO et la communauté Internet ont décrit chacun leur modèle de communication. Les deux organismes de normalisation proposent chacun leur interface d'accès et leur protocole de gestion qui sont, "bien entendu", incompatibles.

Notre analyse est basée sur la même démarche que dans la section précédente : nous décrirons d'abord le modèle de communication de l'ISO pour ensuite analyser celui du monde Internet. Une comparaison des deux modèles précédera la conclusion.

Détailler complètement ces deux modèles pourrait être le seul sujet d'un mémoire. C'est pour cette raison que nous décrirons uniquement ce qui est nécessaire à la compréhension des principes de base, autrement dit les interfaces d'accès. Le lecteur qui désire approfondir ces connaissances sur cette matière pourra consulter les livres référencés dans la bibliographie.

4.3.1 Le modèle de communication de l'ISO

L'ISO distingue clairement, dans son modèle de communication, la notion de service (CMIS[11]) et la notion de protocole (CMIP[12]). Lorsque l'administrateur désire accéder aux informations d'un agent, il utilise l'interface normalisée décrite dans CMIS. La transmission de la requête vers l'agent et la récupération des informations (la réponse) sont prises en charge par le protocole de gestion CMIP. Celui-ci se base sur un protocole de communication pour le transfert de données comme l'illustre la Figure 10 de la page suivante.

[11] CMIS : *Common Management Information Service*, décrit dans la norme [ISO 9595].
[12] CMIP : *Common Management Information Protocol*, décrit dans la norme [ISO 9596-1].

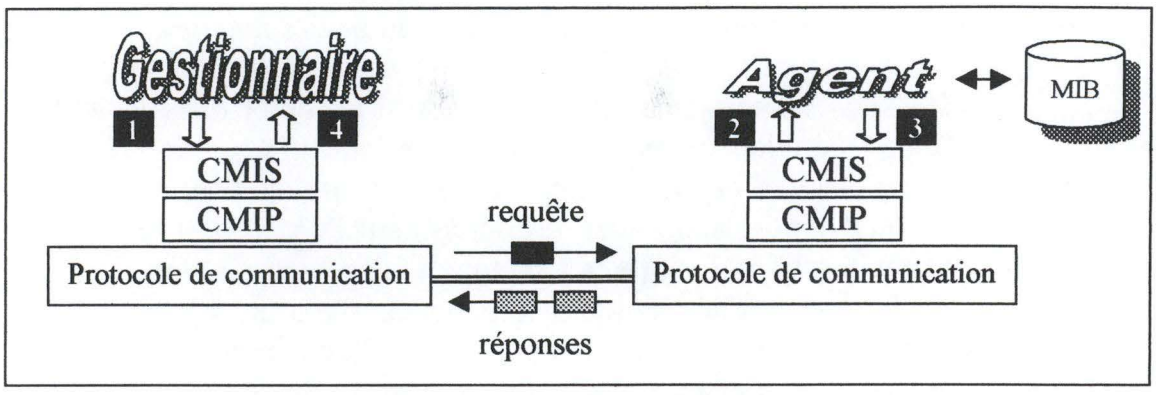


Figure 10 : Le service CMIS et le protocole CMIP

Les interactions de la Figure 10 sont les suivantes :

[1] Le gestionnaire émet une requête en utilisant l'interface de CMIS. Ensuite, cette requête est passée à l'entité protocolaire de gestion CMIP. Cette entité s'occupe de créer les connexions nécessaires à l'échange d'informations. C'est finalement par l'intermédiaire du protocole de communication que la requête est acheminée vers l'agent.

[2] Une fois la requête arrivée à l'agent, elle remonte les différentes couches pour être finalement traitée par la couche CMIS. Celle-ci analyse la requête et collecte les informations souhaitées dans la base d'information de gestion, la MIB.

[3] Etant donné que la taille de la réponse est aléatoire et peut être d'une envergure importante, la couche CMIP gère la fragmentation du côté de l'agent et la défragmentation du côté du gestionnaire. Les différents fragments constituant la réponse sont renvoyés vers le gestionnaire par l'intermédiaire du protocole de communication.

[4] Les fragments sont rassemblés par la couche CMIP pour ne former qu'une seule entité : la réponse. C'est ensuite la couche CMIS qui délivre celle-ci au gestionnaire.

Les différentes requêtes offertes par le service CMIS peuvent être regroupées dans quatre catégories :

- la première est du type « création et suppression » d'un objet,
- la deuxième est du type « mise à jour et lecture » de valeurs d'attributs d'objets,
- la troisième correspond à l'« exécution » d'une action,
- la dernière au « rapport d'événements » pour le service de notification.

Il existe un type d'opération supplémentaire qui correspond à l'annulation d'une lecture.

Les six services d'opérations et le service de notification (ce qui correspond à l'interface du service) sont accessibles par les primitives suivantes :

- M-CREATE pour demander la création d'un objet dans la MIB de l'agent. Cette opération permet en fait à un administrateur de créer une **instance** d'une classe d'objet, ce qu'on a appelé le *managed object (MOI)*.
- M-DELETE pour supprimer des objets dans la MIB de l'agent. Précisons que toutes les opérations (création, suppression, consultation...) sont contrôlées par les services de sécurité incorporés dans le protocole CMIP.
- M-ACTION pour demander l'exécution d'une action sur un ou plusieurs objets. L'administrateur a le choix entre le mode confirmé et le mode non confirmé[13]. Dans le premier mode, l'administrateur reçoit un accusé de réception. Notons que les primitives de création et de suppression sont toujours en mode confirmé.
- M-SET pour modifier les valeurs d'attributs d'objets. Le choix du mode confirmé ou non confirmé est également proposé.
- M-GET pour consulter les valeurs associées aux attributs de l'objet. Cette opération peut être annulée par M-CANCEL-GET. Cette primitive s'exécute toujours en mode confirmé.
- M-EVENT-REPORT pour transmettre un rapport d'événement relatif à un objet. Contrairement aux autres opérations, celle-ci est initialisée par l'agent. Le choix du mode de confirmation est aussi d'application pour cette primitive.

Nous devons, pour conclure notre exposé sur le modèle de communication de l'ISO, encore expliquer les notions indissociables de **portée** (*scoping*) et de **filtre** (*filtering*). Ces deux mécanismes permettent de focaliser la portée des opérations M-GET, M-SET, M-DELETE et M-CREATE par rapport à l'ensemble des objets de gestion présents dans la MIB de l'agent. Autrement dit, si ces deux mécanismes n'existaient pas, les quatre primitives ne pourraient s'appliquer qu'à un seul objet.

Nous avons vu que l'ensemble des objets de gestion contenus dans une MIB était structuré dans un arbre de contenance (*voir page 32*). Les deux mécanismes s'applique sur cet arbre. La Figure 11 illustre le concept de portée. La portée peut concerner un seul objet de gestion (1), un niveau de profondeur particulier (2) ou encore un ensemble de niveaux (3).

[13] Le mode confirmé correspond à celui défini par le modèle OSI : requête – indication – réponse – confirmation. Par contre, le mode non confirmé s'exécute uniquement en deux phases : requête – indication.

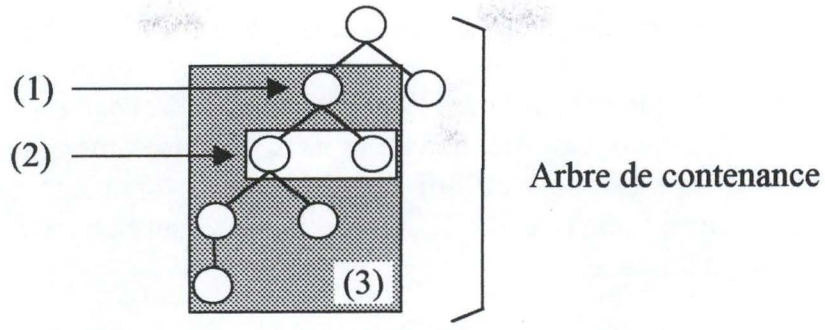


Figure 11 : Le concept de portée

Lorsque la portée porte sur un sous-arbre, il est nécessaire de préciser la racine de celui-ci. Le nœud racine porte le nom dans la terminologie ISO « d'objet de gestion de base » (*base management objet*).

Le filtre est une expression booléenne qui consiste en une ou plusieurs assertions appliquées aux objets de la portée. Chaque assertion est un test d'égalité, d'ordre, de présence ou de comparaison d'ensembles. Un filtre est une combinaison d'assertions basée sur les opérateurs AND, OR ou NOT.

Notons encore que ces deux puissants mécanismes permettent à l'administrateur de récolter l'ensemble des informations qu'il désire en une seule requête, comme par exemple « quels sont les routeurs qui sont actifs et qui possèdent au moins trois interfaces ».

Le Tableau 4 ci-dessous résume les six opérations et la notification ainsi que leur propriété.

Opération (notification)	**Service**	**Mode**	**Filtre et portée**
Création	M-CREATE	Obligatoirement confirmé	Oui
Destruction	M-DELETE	Obligatoirement confirmé	Oui
Action	M-ACTION	Confirmé ou non	Non
Mise à jour	M-SET	Confirmé ou non	Oui
Consultation	M-GET M-CANCEL-GET	Obligatoirement confirmé	Oui *Pas de sens*
Notification	M-EVENT-REPORT	Confirmé ou non	*Pas de sens*

Tableau 4 : Opérations du service CMIS

4.3.2 Le modèle de communication de la communauté Internet

SNMP (*Simple Network Management Protocol*) est né dans la communauté Internet des milieux de la recherche et de la défense américains. Défini dans le [RFC-1157], SNMP s'est aujourd'hui imposé comme standard de fait pour les réseaux TCP/IP. Contrairement à l'ISO, les notions de service et de protocole ne sont pas clairement distinguées.

Le protocole de gestion SNMP se situe, dans l'architecture protocolaire du DoD, au niveau de la couche application (voir Figure 12). Il se base sur le protocole de transport en mode non connecté UDP. [VANB 95] explique ce choix : « *En effet, devant pouvoir être utilisé sur tous les types de réseaux et particulièrement dans les cas de situations critiques où le réseau est surchargé (afin de détecter éventuellement la cause du problème), ce protocole ne peut exiger un transfert de données fiable exigeant une ouverture de connexion préalable et des retransmissions nombreuses* ».

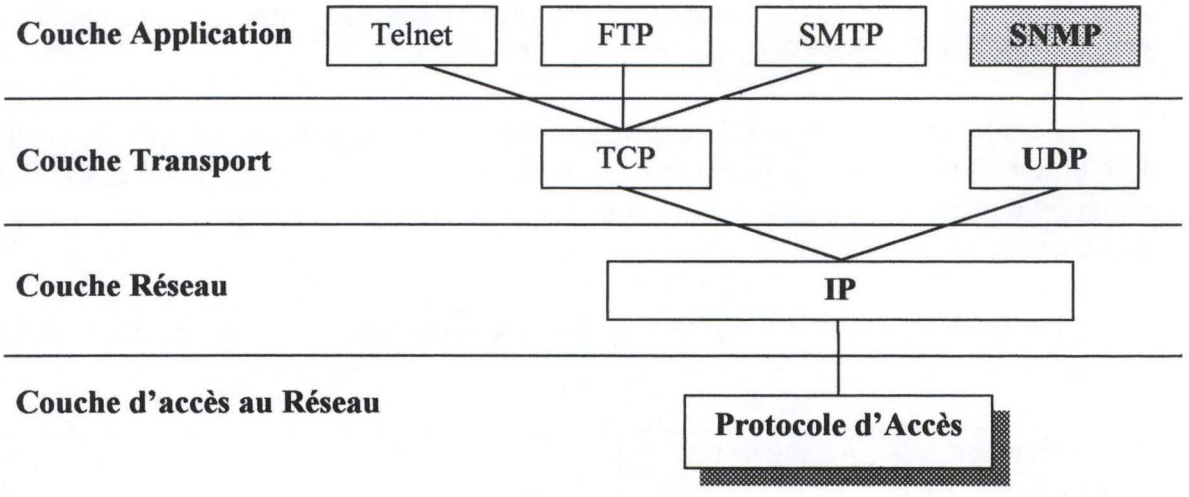


Figure 12 : Architecture protocolaire TCP/IP

Les opérations de gestion sont architecturées en transactions « question-réponse » entre le gestionnaire et les agents. Le protocole SNMP possède cinq types de messages :

- trois requêtes : *GetRequest*, *GetNextRequest* et *SetRequest* ;
- une réponse : *GetResponse* ;
- un message d'événement : *Trap*.

La requête *GetRequest* émise par le gestionnaire est analysée par l'agent qui consulte, dans la MIB, les objets précisés en argument. L'agent répond au gestionnaire par l'envoi d'une primitive *GetResponse* contenant la valeur des objets demandés.

GetNextRequest est utilisée pour les lectures séquentielles d'informations dans la MIB : cette commande est particulièrement utilisée pour la lecture des tables dans la MIB. Après avoir lu un premier enregistrement de la table avec la requête *GetRequest*, les autres enregistrements de la table sont lus de manière séquentielle par une série de *GetNextRequest*.

La requête *SetRequest* est utilisée par le gestionnaire pour effectuer des modifications dans la MIB : à la réception de cette commande, l'agent met à jour les variables de la MIB à partir des valeurs passées en argument. Chacune des variables doit être précisément indiquée, et la valeur doit être en accord avec la syntaxe de la variable à modifier, sinon l'agent signale une erreur. *SetRequest* est une commande puissante, car sa mauvaise utilisation peut altérer des paramètres importants influant sur le fonctionnement correct du réseau.

Pour chaque requête du gestionnaire (lecture et modification), l'agent répond en utilisant *GetReponse*. Cela peut être une réponse positive (exécutant ou confirmant l'accomplissement de l'opération demandée) ou négative dans le cas d'erreurs.

Trap est une commande spéciale qui est émise par l'agent vers le gestionnaire lors d'un événement particulier, spécifié *a priori*. Le protocole SNMP prévoit six événements qui correspondent chacun à un numéro :

- *coldStart* (0) : démarrage à froid ;
- *warmStart* (1) : démarrage à chaud ;
- *linkDown* (2) : détection d'une chute d'un lien de communication avec la valeur de *ifIndex* pour identifier l'interface affectée ;
- *linkUp* (3) : détection de la mise en route d'un lien de communication ;
- *authenticationFailure* (4) : refus d'authentification d'un message SNMP ;
- *egpNeighborLoss* (5) : perte de partenaire EGP.

Le protocole permet toutefois de définir d'autres *traps* signalées par la valeur *enterpriseSpecific* (6). Un autre champ de l'alarme permet de spécifier une deuxième valeur pour identifier la cause de l'événement.

Nous ne détaillerons pas le format des messages SNMP, mais nous voulons souligner que la seule notion de sécurité réside dans le concept de communauté. Tous les messages émis par le gestionnaire contiennent le champ *community*. Celui-ci est une chaîne de caractères censée identifier la provenance du message, et donc informer l'agent SNMP sur les opérations permises. Cette chaîne correspond en quelque sorte à un mot de passe écrit **en clair** dans le message. Autrement dit, les messages n'étant pas cryptés, un intervenant extérieur qui réussirait à écouter sur le média de transport les messages SNMP, serait parfaitement en mesure d'identifier les différentes

opérations soumises par le gestionnaire aux agents SNMP. Il pourrait même modifier certaines valeurs du message et, plus grave encore, usurper l'identité de l'administrateur pour accéder à d'autres agents.

Evidement, les responsables du protocole SNMP étaient conscients que la sécurité était plus que rudimentaire. C'est pour cette raison qu'ils ont créé une deuxième version du protocole : SNMPv2. Cette version, en plus d'améliorer l'interface, enrichit l'aspect sécurité et fournit des messages authentifiés et encryptés sur la base de l'algorithme DES (*Data Encryption Standard*) . La description complète de SNMPv2 se trouve dans le [RFC-1448]. Signalons que les concepteurs de la communauté Internet travaillent, à l'heure actuelle, sur une troisième version du protocole : SNMPv3.

Nous avons vu, dans l'étude du modèle informationnel, que la communauté Internet avait spécifié un RFC décrivant la MIB dans le langage de l'ISO. Cette migration vers le modèle ISO est aussi d'application pour le protocole de gestion SNMP. Pour cela, un groupe de travail appelé OIM (*Osi Internet Management*) a été créé au sein de l'ISO (et plus particulièrement au sein de l'OSI qui s'occupe de la normalisation des réseaux). Ce groupe travaille sur CMOT (*CMip Over Tcp/ip*) ; son principal rôle est de produire un ensemble de spécifications qui offre les services CMIS et le protocole CMIP sur des réseaux de type TCP/IP. L'architecture envisagée pour CMOT consiste à rajouter au-dessus des protocoles TCP et UDP une couche présentation simplifiée (LPP, *Lightweight Presentation Protocol*) ainsi que les différents services comme CMIS et CMIP. Cependant, bien que la démarche semble séduisante, il n'y a pas, à l'heure actuelle, de véritable implémentation de ce protocole.

4.3.3 CMIP vs SNMP

Rappelons que SNMP est un standard de fait pour les systèmes basés sur le protocole TCP/IP, alors que CMIP est standardisé par l'ISO.

Nous constatons que sur beaucoup de points, SNMP et CMIP semblent très proches, mais, en fait, ils sont fondamentalement différents :

- ✓ SNMP ne rapporte que l'information élémentaire demandée. Il peut fournir une suite d'attributs à condition de les demander explicitement. De plus, la consultation d'une seule table demande l'émission de plusieurs requêtes par le gestionnaire. Par contre, avec CMIP, une seule requête désignant l'objet convoité (voire le sous-arbre) suffit pour récupérer l'ensemble des informations.
- ✓ CMIP est un protocole orienté connexion, contrairement à SNMP qui est sans connexion (UDP). Si cette dernière solution a l'avantage de minimiser

le trafic réseau, elle ne permet pas de garantir que l'agent a reçu la requête, inconvénient d'autant plus important lorsqu'il s'agit d'une modification.

✓ La surveillance d'un agent par un gestionnaire s'effectue généralement de deux manières différentes : soit par *polling* (cas de SNMP), soit par *reporting* (cas de CMIP). Le *polling* consiste à interroger l'agent à intervalles réguliers sur son état (sur la valeur de certaines variables)[14]. Par contre, le *reporting* permet à l'agent, en fonction des assertions sur les attributs de l'objet et les valeurs courantes de ceux-ci, d'émettre une notification. Si la technique de *polling* présente l'avantage de renforcer la sécurité du système, elle tend cependant à générer beaucoup de trafic sur le réseau.

Le Tableau 5 ci-dessous résume les différences entre les deux protocoles.

CMIP/CMIS	SNMP
Mode connecté	Mode non connecté
Interface Objet	Interface Attribut
Possibilité de scoping et de filtering	Néant
Gestion par événement (M-EVENT-REPORT)	Gestion par polling (événements limités possibles : TRAP)
Actions particulières possibles sur les objets (M-ACTION, M-CREATE, M-DELETE)	Pas de possibilités d'actions particulières

Tableau 5 : Comparaison CMIP/SNMP

4.3.4 Conclusion

Nous avons vu dans ce chapitre quels étaient pour l'administrateur les différents moyens d'accéder aux informations gérées par les agents. Chacun des deux modèles de communication propose leur propre interface. Nous avons remarqué aussi que CMIP était plus puissant et plus complet que SNMP. Notons que cet avantage pose le problème de ne pas pouvoir être présent sur tous les équipements (dû à sa consommation en mémoire). Les évolutions futures permettront une intégration du protocole CMIP dans le réseau TCP/IP de la communauté Internet.

[14] Rappelons que le concept de TRAP dans SNMP permet à l'agent d'envoyer une alarme au gestionnaire mais que le déclenchement de celle-ci ne peut pas s'effectuer par rapport à la valeur d'une variable (contrairement à CMIP).

Chapitre 5 : Conclusion de la première partie

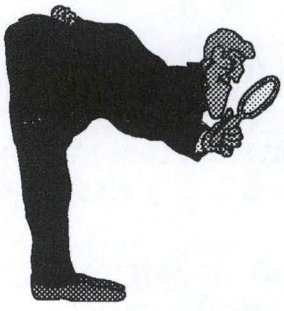
Le monde de l'administration s'intègre dans un environnement complexe et hétérogène. Le « qu'administre-t-on » nous a permis d'identifier les composants qui étaient pris en charge par l'administration. Nous avons vu aussi que les fonctionnalités de l'administration étaient nombreuses et diverses.

Ensuite, le « pourquoi administrer » a permis de dégager les enjeux et les finalités de l'administration. La notion importante de qualité de service a été définie. C'est par l'intermédiaire de quatre critères génériques que nous avons pu évaluer et quantifier la qualité de service d'un réseau. L'évolution des réseaux de communication, par l'intégration d'une interface unique les représentant une seule entité atomique et homogène, a été un point fondamental abordé dans cette partie.

Pour le « comment administrer », deux modèles se sont révélés importants. Le modèle informationnel a été nécessaire pour comprendre comment un équipement ou un service était perçu par l'administration. Les deux standards (*de facto* et *de jure*), mondialement répandus, ainsi que leur comparaison ont été présentés. Le modèle de communication permet de centraliser l'administration. Il est responsable de la récolte, de manière la plus transparente possible, des informations nécessaires à la vue globale du réseau. L'étude des deux modèles nous a permis de comprendre quels types d'informations étaient récoltés et quels en étaient les moyens d'accès (lecture, modification, etc.).

Seconde Partie

LES OUTILS DE L'ADMINISTRATION



Les sujets abordés sont :

- Les plates-formes d'administration (avec l'étude d'ISM et d'OPENVIEW)
- La distribution de logiciels

Chapitre 6 : Les plates-formes d'administration

6.1 Introduction

Lorsque nous avons présenté le modèle architectural de l'administration, nous avons dégagé deux entités, l'agent et le gestionnaire, assumant chacune un rôle spécifique. Chaque composant (équipement et service) du réseau est perçu par l'administration à travers un agent. Celui-ci est responsable de sa MIB, dépositaire conceptuel des informations de gestion. Le gestionnaire a le rôle essentiel de centraliser toutes les informations issues du réseau, de recevoir, de traiter et de stocker les alarmes émises par les agents ; autrement dit, son rôle est de créer la vue globale du système.

A cause de sa complexité fonctionnelle, le gestionnaire possède une architecture compliquée et se présente sous la forme d'un ensemble de processus coopérants. De plus, le gestionnaire doit offrir les moyens de spécialiser les services de l'administration en fonction des besoins spécifiques du client ; il doit donc être générique et évolutif. C'est pour cette raison que le gestionnaire est considéré plus comme une boite à outils que comme une seule application. Cette boite à outils porte le nom, dans le jargon de l'administration, de **plate-forme d'administration**.

Les fonctionnalités de base offertes par les plates-formes sont nombreuses :

- la découverte automatique de la topologie du réseau (ponts, routeurs, réseaux locaux, liaisons longue distance, stations de travail, etc.) ;
- la compilation de nouvelles descriptions de MIB, l'interrogation d'agents à travers un outil appelé *MIB Browser* et la modification des informations stockées dans les MIB ;
- l'enregistrement de toutes les alarmes (notifications) qui sont générées par les équipements réseaux ;
- l'affichage, via une interface graphique conviviale, des topologies du réseau à travers différentes cartes, l'indication visuelle de l'état des composants (à l'aide de couleurs, par exemple) ;
- le développement d'outils complémentaires et la création d'applications spécifiques aux clients via une interface de programmation.

Remarquons qu'une plate-forme d'administration, pour fonctionner correctement, doit reposer sur un système d'exploitation multitâche afin d'être capable de traiter plusieurs événements simultanés : déplacement de fenêtre par l'administrateur, réception d'une ou plusieurs alarmes, interrogation des agents, etc.

Comme nous l'avons fait remarquer, la plate-forme est une boîte à outils génériques ; autrement dit, c'est une base de départ sur laquelle il faut réaliser un

important travail de paramétrage et de personnalisation, ainsi que des développements supplémentaires.

Cinq plates-formes se partagent le marché de l'administration réseau, toutes basées sur un système Unix[15] :

- **Sun Net Manager** (SNM en abrégé) sur système Solaris (un Unix système V développé par la société Sun pour ses machines) ;
- **HP Open View** (HPOV en abrégé) de Hewlett-Packard sur système HP/UX (un Unix système V développé par HP) ou sur système Solaris ;
- **NetView/6000** (NW/600 en abrégé) sur système AIX (un Unix système V développé par IBM pour les machines RISC/6000) ou sur système Solaris ;
- **Sprectrum** développé par la société Cabletron et, historiquement, le premier outil de ce type ;
- **Integrated System Management** (ISM en abrégé) développé par la société BULL.

Le fait qu'il y ait peu de concurrents sur ce marché témoigne de la complexité de ces outils et donc de leur coût de développement. Les autres plates-formes sont dérivées des trois premières précédemment citées. Par exemple, Polycenter de la société DEC, qui fonctionne sur machine Alpha, est directement issu de NetView/6000, lui-même construit sur les principes d'HP OpenView.

A une plus petite échelle, et pour répondre à des besoins plus restreints, certains systèmes sont présents sur le marché des plates-formes sous PC-Windows. De nombreux systèmes existent, mais peu peuvent prétendre au titre de plate-forme, c'est-à-dire de base logicielle sur laquelle des développements peuvent être réalisés ; citons, par exemple, le produit Netware Management System (NMS en abrégé) de la société Novell.

Voici la démarche que nous avons adoptée pour ce chapitre : dans un premier temps, nous proposerons des critères pour l'évaluation des plates-formes ; ensuite, nous détaillerons les deux plates-formes *ISM* (qui fera l'objet d'une étude plus poussée) et *HP OpenView* ; finalement, avant de conclure, nous aborderons brièvement le concept d'agent intelligent.

[15] En raison de l'importance que prend Windows NT à l'heure actuelle, les constructeurs de plates-formes commencent à porter leur produit sur ce système d'exploitation multitâche.

6.2 Les critères de choix

Le choix d'une plate-forme d'administration est une opération délicate qui doit tenir compte de la stratégie globale ou informatique de la société. En effet, le coût d'une plate-forme peut avoisiner les quelques millions de francs belges (comprenant le logiciel et le matériel) ; viennent s'ajouter en plus les coûts de développement des applications spécifiques.

Les apports supposés d'une plate-forme sont :

- ✓ une réduction des temps et coûts de développement grâce à la boite à outils ; la mise à disposition de services et de composants applicatifs génériques donne la possibilité de prendre en compte rapidement les besoins spécifiques des clients ;
- ✓ une harmonisation des Interfaces Homme-Machine (IHM) ;
- ✓ une intégration du modèle informationnel qui induit une visibilité homogène des ressources et qui permet de réduire le temps d'apprentissage ;
- ✓ une optimisation et une spécialisation de l'exécutif ; l'optimisation permet de garantir de bonnes performances tandis que la spécialisation permet de rendre transparents les différents protocoles de gestion utilisés.

Par rapport à ces attentes, on peut déduire les points à comparer et les questions qu'on doit se poser lors du choix de la plate-forme. Ensuite, les administrateurs doivent pondérer en fonction de leur attente et de leur environnement. Une grille d'évaluation permet d'avoir une estimation globale.

Si les critères tels que les possibilités offertes par la plate-forme, sa portabilité, sa flexibilité et la convivialité de son interface sont déterminants, il ne faut pas sous-estimer la performance. En effet, un volume de trafic réseau trop important, généré par la plate-forme, peut considérablement diminuer les performances globales du réseau[16]. Autrement dit, l'administrateur devra tenir compte, en plus des possibilités offertes de la plate-forme, de ses performances.

[16] Rappelons qu'un des buts fondamentaux de l'administration est de garantir un bon niveau de fonctionnement du réseau, autrement dit, de maintenir de bonnes performances.

6.3 La plate-forme ISM de Bull

La gestion intégrée et la sécurité des systèmes sont des éléments essentiels du modèle informatique réparti de Bull. Afin de placer ISM dans son contexte, il est nécessaire de présenter ce modèle, appelé **DCM** (*Distributed Computing Model*).

6.3.1 Le modèle DCM

DCM se compose essentiellement du DCF (*Distributed Computing Facilities*), lui-même s'appuyant sur le DCE (*Distributed Computing Environnement*) défini par l'OSF[17]. Le DCF permet l'interconnexion de systèmes d'exploitation hétérogènes.

Le DCE, pour sa part, est le modèle de programmation client/serveur qui a été adopté entre autres par Bull, IBM et HP.

Le modèle DCM a pour objectif de mettre l'utilisateur au cœur du système d'information. Il répond à trois types de fonctions :

- *Fonction utilisateur* : offrir des applicatifs sectoriels (banque, assurance, industrie, distribution, secteur public, etc.) répondant aux besoins des entreprises et des administrations.
- *Fonction développeur* : mettre en œuvre un environnement de développement qui exploite le capital informationnel et réduit les coûts d'implantation.
- *Fonction administrateur* : fournir des outils d'administration et de sécurité pour contrôler et superviser les ressources systèmes et interdire la violation des données.

L'ouverture du modèle impose la disponibilité d'interfaces pour chaque service au sein de chaque composant. Les interfaces des services sont documentées et elles reposent sur des normes *de facto* ou *de jure*. Ceci permet de disposer d'une "interopérabilité" large, de faciliter la portabilité et/ou l'intégration des logiciels.

La caractéristique essentielle qui permet à ce modèle d'offrir un nouveau niveau de convivialité est sa transparence. L'utilisateur dispose d'un accès transparent aux applications qui peuvent résider sur de multiples systèmes d'exploitation, les applications peuvent faire usage des services distribués dans le réseau.

---

[17] L'OSF (*Open Software Foundation*).est un consortium d'entreprises qui travaillent ensemble pour développer des logiciels destinés au marché des systèmes ouverts. L'OSF décide d'implémenter une technologie et invite ses membres à soumettre des propositions pour le développement du logiciel. La (ou les) entreprise(s) élue(s) développe(nt) le logiciel, l'OSF assurant la conduite du projet, et elle(s) apporte(nt) la technologie à l'OSF. A son tour, celle-ci fournit le code source (logiciel et documentation) à toutes les entreprises qui ont acquis une licence du logiciel DCE, [ROSEN et al 93].

Le DCM est représenté comme un ensemble de composants dont chacun décrit une famille de services qui ont un rôle fonctionnel commun. Graphiquement, le modèle est représenté en trois dimensions. Il est constitué de quatre composants horizontaux qui sont eux-mêmes encadrés par deux composants verticaux (voir Figure 13). La troisième dimension représente les applications sectorielles.

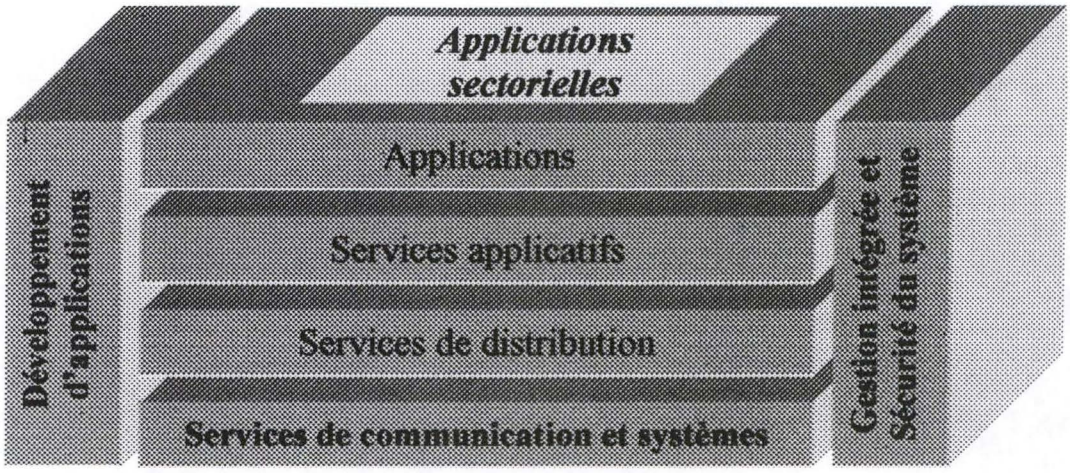


Figure 13 : Les composants du modèle DCM

Le composant **Applications** représente les applicatifs dont l'utilisateur final peut se servir pour atteindre les objectifs de son entreprise. Ce composant contient deux catégories d'applicatifs : les applicatifs génériques (par exemple la bureautique, ou le traitement de l'image), et les applicatifs spécifiques qui sont adaptés aux besoins et exigences d'un secteur d'activité économique particulier (banque, assurance, etc.).

Le composant **Services applicatifs** comporte des services appelés par les applications et qui se répartissent en trois catégories, *VOIR*, *ECHANGER* et *TRAITER*, selon leur principal domaine d'efficacité. Les services classiques comme la présentation ou l'impression, ne sont plus nécessairement exécutés sur le système local. Par exemple, un applicatif peut effectuer une présentation graphique sur une station, l'impression sur une imprimante locale ou distante présentant les caractéristiques appropriées et la facturation sur un serveur spécialisé. Les utilisateurs n'ont donc plus à se préoccuper de l'endroit où les services sont effectivement rendus ; ils ne doivent pas non plus connaître les chemins logiques ou physiques que les données doivent parcourir dans le réseau.

Le composant **Services de distribution** offre des services d'appel à distance (RPC : *Remote Procedure Call*), d'annuaire et d'horloge. Ces services assurent la transparence de la localisation des services applicatifs. L'appellation des objets appartenant au système distribué est logiquement identique.

Le composant **Service de communication et systèmes** contient les services qui assurent le transport effectif des informations et qui fournissent la puissance de calcul du système. Les protocoles OSI et TCP/IP sont, entre autre, supportés.

Le composant **Développement d'applications** comporte un certain nombre d'outils et de services dont doit disposer le développeur d'applications. Celui-ci a accès à toutes les interfaces ouvertes proposées par le modèle DCM afin de construire des solutions dans un environnement distribué.

Le but du modèle est d'offrir à l'utilisateur un accès à des ressources distribuées sur différents systèmes d'exploitation et ce, de manière totalement transparente. Il est clair qu'un tel système doit être géré et contrôlé afin notamment de garantir la sécurité. Dans cet esprit, Bull a décidé d'ajouter un composant d'administration et de supervision des différents éléments du modèle DCM. Ce composant est ISM (appelé dans le modèle *gestion intégrée et sécurité du système*), comme l'illustre la Figure 14.

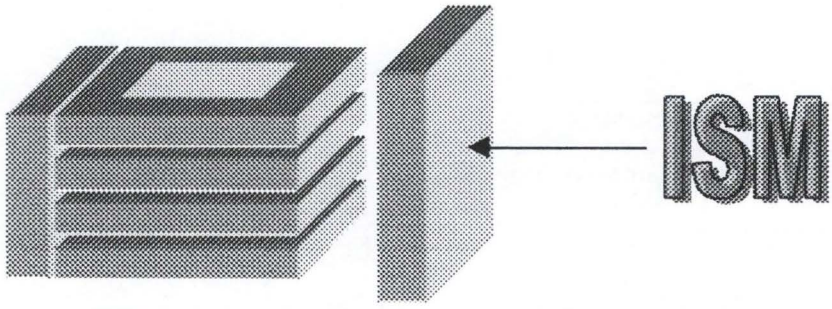


Figure 14 : ISM, brique architecturale verticale du modèle

Maintenant que nous avons replacé ISM dans son contexte, nous pouvons analyser son architecture et l'ensemble des services qu'il propose.

6.3.2 L'architecture d'ISM

ISM décompose le problème d'administration en trois composants de base :

- les **Services** : ceux-ci sont découpés en Services de Gestion et Services de Communication et représentent ensemble le cœur de la plate-forme ;
- les **Intégrateurs** : il existe des Intégrateurs de Gestionnaire et des Intégrateurs d'Agents ;
- les **Applications** : celles-ci sont nombreuses et offrent chacune un service particulier que nous décrirons ultérieurement.

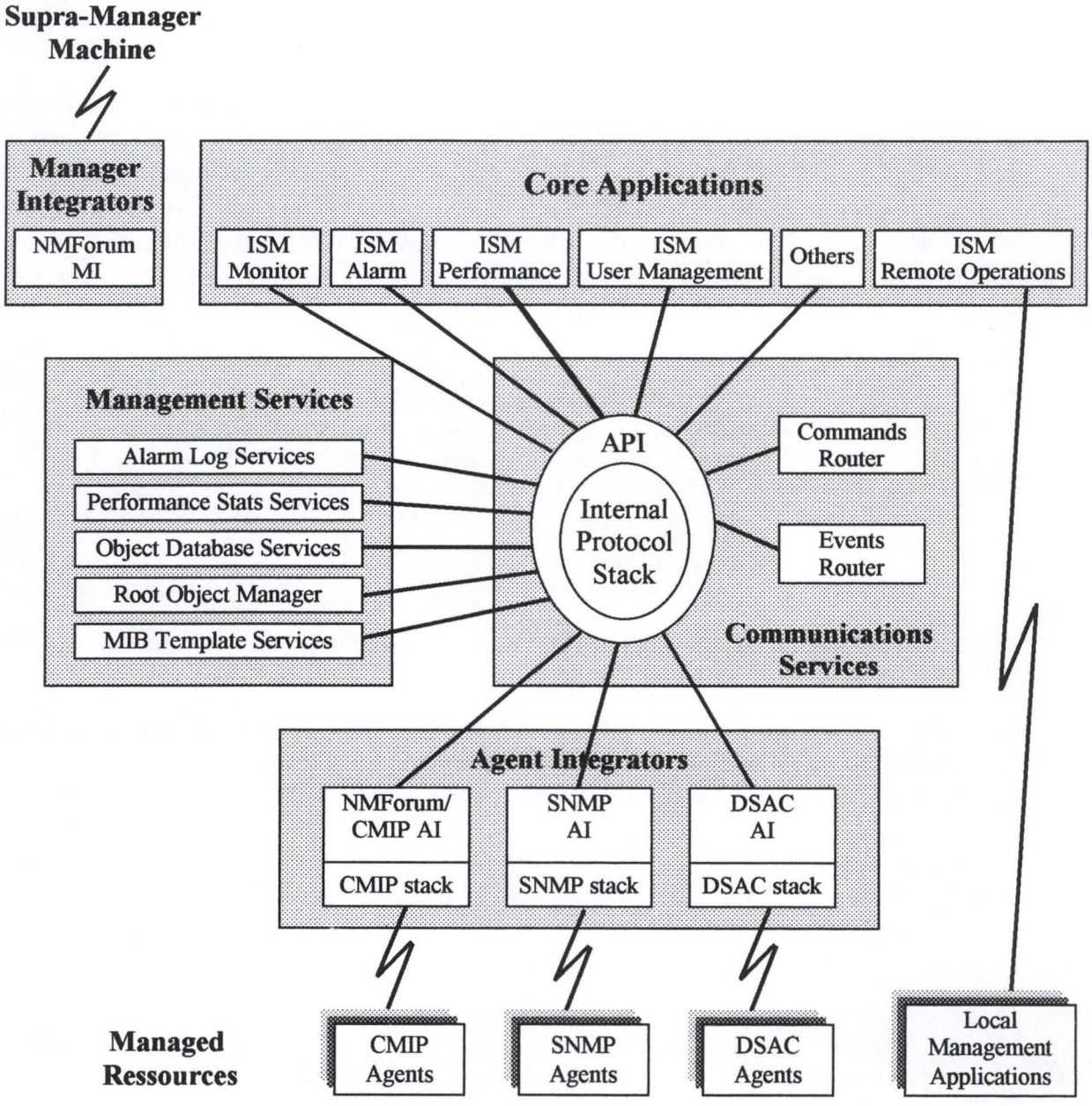


Figure 15 : L'architecture d'ISM

ISM est une plate-forme **orientée objet** qui a été conçue conformément aux **normes et aux standards** existants. Elle est adaptée à une large étendue de composants réseaux et est en accord avec les définitions de l'OSF. ISM supporte trois protocoles de gestion standard :

- ISO/CMIP : cette architecture utilise CMIS/CMIP, standard préféré d'ISM. Le protocole CMIP est d'ailleurs celui qui a été choisi pour les communications internes à ISM comme interface de communication au niveau de l'interconnexion entre l'administrateur et d'autres plates-formes, dans la mesure où celles-ci supportent le protocole NMForum ;
- SNMP : le standard *de facto* ;
- DSAC : DSAC (*Distributed Systems Administration and Control*) est utilisé pour l'administration de réseaux OSI/DSA.

Néanmoins, chaque protocole (CMIP, SNMP et DSAC) possède ses particularités et son propre format de requêtes. Le but, au sein d'ISM, est de formuler les requêtes de manière unique et ce, indépendamment des agents à interroger. Cependant, ces agents ne comprennent que leur « langage » et il faut nécessairement ajouter des composants capables de comprendre les requêtes ISM et de les reformuler en fonction des agents qui sont sollicités. Ce sont ces composants qui sont appelés *intégrateurs d'agents* (*AI*) ou en anglais, *agent integrators* (voir Figure 15). Notons que ceux-ci sont une spécificité d'ISM. Les AI sont des *drivers* d'agents, de la même manière qu'il existe des *drivers* d'imprimantes. Ils supportent les services standards d'ISM : CREATE, DELETE, GET, SET, ALARM et ENROL ET ils possèdent aussi les capacités de *scoping* et de *filtering* qui, rappelons-le, sont inexistantes dans le protocole SNMP.

Les intégrateurs de managers (MI) permettent à une plate-forme de se comporter comme un agent sous le contrôle d'un supramanager. Avec ce mécanisme, il est possible de créer une hiérarchie de plates-formes et de déléguer l'administration par domaine, le supramanager étant le coordinateur.

Les composants d'ISM (applications, services, AI) communiquent entre eux en utilisant l'*infrastructure de communication*. Elle fournit les fonctions de transfert de messages d'ISM et supporte la répartition des différents composants à travers le système distribué. Elle comprend entre autre une interface de programmation (API) basée sur CMIS, une pile de protocoles internes pour les communications entre différentes plates-formes ISM, un routeur de commandes et un routeur d'événements.

Le routeur de commandes (*commands router*) permet de rendre transparente la localisation physique de la ressource (ainsi que son agent). Les requêtes sont ainsi routées vers les agents auxquels elles sont destinées. Pour cela, la requête doit être analysée et une table de routage interne utilisée pour le routeur.

Le routeur d'événements (*events routeur*) permet à une application de la plate-forme de « souscrire » à un événement. En effet, les événements ne sont pas envoyés à toutes les applications mais seulement à celles qui en font la demande au préalable. *Alarm Log Services* utilise ce service pour filtrer les notifications d'alarmes provenant des AI.

Les services d'ISM ont été créés pour faciliter le développement des applications. Ils permettent de stocker les alarmes envoyées par les agents, d'établir des statistiques et de générer des alarmes en cas de dépassement de seuils, de stocker les formulaires (*templates*) des différents objets du système, etc. L'utilisation de ces services se fait aussi par l'utilisation de l'interface CMIS.

L'ensemble des applications d'ISM forme un noyau central. Celui-ci est constitué d'applications génériques et d'applications complémentaires. Autour de ce noyau ont été créés six domaines applicatifs (appelés aussi modules). En plus des applications spécifiques, les domaines regroupent les applicatifs nécessaires du noyau pour constituer un « bloc de gestion ». Cette conception modulaire permet à un client d'acquérir uniquement les applications dont il a besoin. L'architecture complète est appelée ISM/OpenMaster (voir Figure 16).

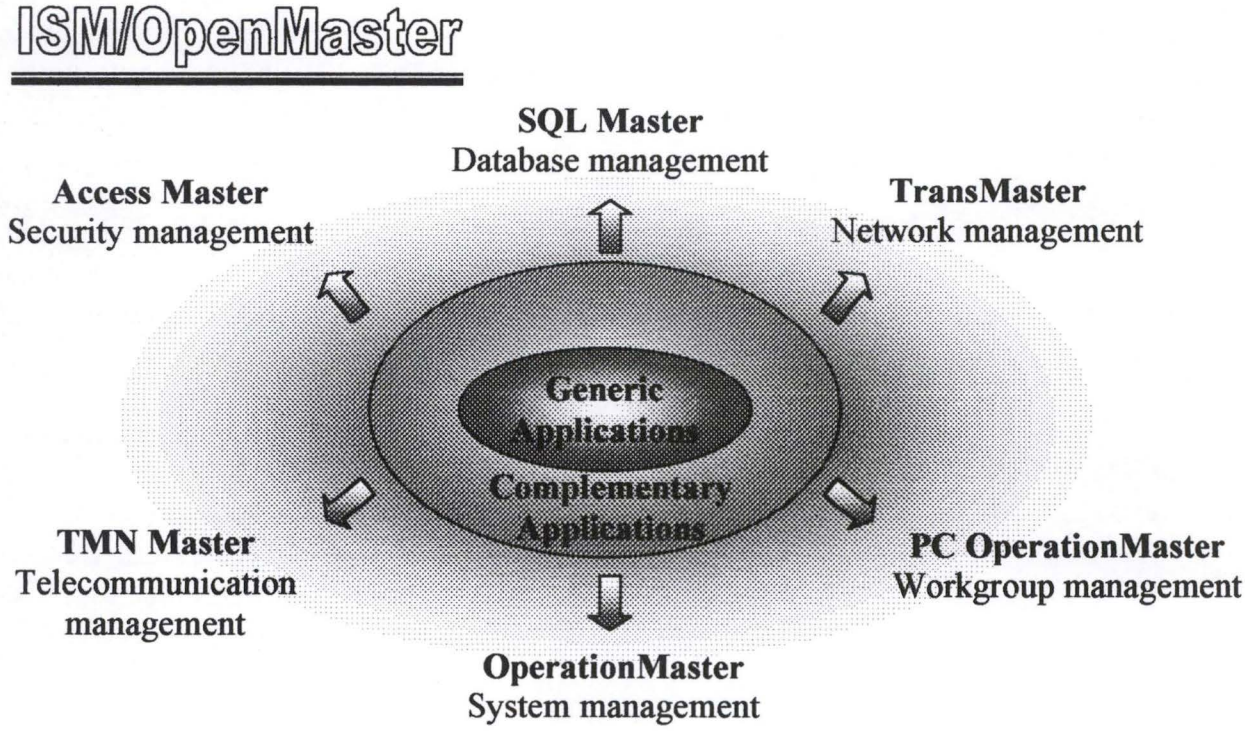


Figure 16 : Conception modulaire des applications

- ISM AccessMaster permet aux utilisateurs finaux de voir le système hétérogène comme une seule entité sécurisée. Basés sur une identification et

une authentification uniques (autrement dit, un seul « login »), l'utilisateur connecté au système dispose automatiquement de l'ensemble des applications réparties auxquelles il peut accéder. De plus, ISM AccessMaster autorise les administrateurs à paramétrer et à contrôler la sécurité du système ; par exemple, ils peuvent aisément modifier les droits d'accès des ressources. De plus, ils ont la possibilité de surveiller constamment le système avec un niveau de détails très élevé (qui accède à quelle ressource ?, quand cet accès a-t-il été effectué ?, qui est connecté au système ?, etc.).

- ISM SQLMaster fournit un environnement intégré et convivial pour gérer les bases de données distribuées ORACLE, SYBASE, INFORMIX et MICROSOFT SQL Server.
- ISM TransMaster (IT) est capable de gérer plusieurs réseaux hétérogènes en utilisant des protocoles tels que SNMP, CMIP et DSAC. Le résultat est une vue administrative du réseau homogène et atomique. Cette vue homogène permet de gérer les fautes, les configurations et d'évaluer les performances efficacement.
- ISM PC OperationMaster fournit une gestion cohérente des groupes de travail (*workgroups*) basés sur les serveurs Windows NT[18] de Microsoft, NetWare de Novell ainsi que les serveurs LanManger. Les administrateurs sont capables de consulter les inventaires matériels et logiciels des stations de travail et des serveurs. Ils peuvent aussi surveiller l'état des différentes machines (indépendamment du système d'exploitation) : utilisateurs connectés, utilisations des ressources, etc. Ils ont aussi la possibilité de distribuer des logiciels sur des PC avec une utilisation minimale du réseau[19].
- ISM OperationMaster a été spécialement conçu pour la surveillance des performances des machines. A l'aide des outils fournis, les administrateurs sont capables de définir des scénarios qui, en fonction d'un événement, effectueront automatiquement les actions appropriées.
- ISM TMNMaster a été développé particulièrement pour les opérateurs de télécommunication et les fournisseurs d'accès (*providers*). Ceux-ci l'utilisent pour la gestion des réseaux, des services et des différentes ressources dans un environnement hétérogène et distribué. ISM TMNMaster comprend aussi des kits de développement pour améliorer, étendre et personnaliser rapidement les services.

Avant de nous intéresser aux détails de certaines applications fournies par ISM, il est nécessaire de préciser dans quel environnement elles vont évoluer et quelle sera l'interface entre ces applications et les utilisateurs.

L'interface utilisateur d'ISM s'appuie sur MOTIF et X/Windows pour fournir un environnement multifenêtré. Chaque fenêtre est une vue d'une tâche de gestion

[18] Windows NT (*Windows New Technology*)
[19] La distribution de logiciel sera approfondie dans le chapitre 7.

disponible au travers de l'application. Les fenêtres peuvent agir indépendamment les unes des autres, chacune d'elles étant mise à jour à la réception de nouvelles informations en provenance du système (des agents par exemple). ISM tire profit de la puissance du multi-fenétrage grâce à l'exécution des requêtes en mode asynchrone.

L'apparence et les principes d'utilisation de l'interface utilisateur d'ISM sont les mêmes pour toutes les applications. Pour cela, ISM s'appuie sur trois composants de base :

- ISM Monitor : cette application générique contient les fonctions du plus haut niveau de l'interface. Les objets sont représentés à l'écran par des icônes ou par des tableaux ;
- un mécanisme permet l'affichage de toutes les conditions d'exception sur les objets. L'application ISM Alarm est utilisée pour afficher et gérer toutes ces alarmes.
- un langage pour l'administration, SML (*ISM Management Language*), permet la création de nouvelles applications. Il s'agit d'un langage de très haut niveau supportant l'interface utilisateur d'ISM.

Revenons maintenant aux deux types d'applications d'ISM. Les applications génériques sont en mesure de gérer tous les objets de gestion. Elles comprennent leur syntaxe mais pas leur sémantique. Elles fournissent un ensemble de fonctions de base indépendamment du sous-système géré. Par contre, les applications spécifiques agissent uniquement sur certains objets et comprennent la sémantique de ceux-ci. Les fonctions proposées par ce type d'applications sont spécifiques aux objets qu'elles gèrent. Le Tableau 6 ci-dessous reprend la liste des applications génériques les plus importantes d'ISM ainsi qu'une brève description.

Application générique	Description
ISM Monitor	Permet d'afficher des cartes graphiques avec des possibilités d'animations. Cette application est la plus générique et la plus utilisée d'ISM.
ISM Discovery	Permet d'explorer périodiquement le réseau et d'afficher, grâce à ISM Monitor, une carte avec les équipements trouvés.
ISM Alarm	C'est le gestionnaire d'alarmes pour les systèmes et les réseaux. Les alarmes sont affichées (dans ISM Monitor) en temps réel et un historique est conservé pour chaque alarme.
ISM Performance	C'est une application qui est utilisée pour afficher en temps réel des indicateurs de performances, pour définir des seuils d'erreurs pour la génération d'alarmes et les statistiques, et pour calculer et afficher des indicateurs de qualité de service.

Tableau 6 : Les applications génériques essentielles

6.3.3 Conclusion

De nos jours, ISM est considéré comme une des cinq meilleures plates-formes d'administration du marché [BROWN 97]. Son respect pour les normes ainsi que la multiplicité des protocoles et des systèmes d'exploitation supportés permettent son déploiement sur la plupart des réseaux hétérogènes. Grâce à son architecture modulaire et ses outils de développement, les clients peuvent configurer la plate-forme ISM pour qu'elle réponde au mieux à leurs besoins. Le coût est ainsi adapté à chaque client.

Nous terminons notre étude d'ISM par l'appréciation d'un client, extraite d'une revue informatique spécialisée [RESEAUX-662 96] :

« une plate-forme ISM fonctionne depuis février 1995 aux caisses de prévoyance et de retraite de la SNCF. Cette institution gère les dossiers d'un million de bénéficiaires de l'assurance maladie et de 348 900 pensionnés. Au total, près de 150.000 transactions quotidiennes sont traitées, en temps réel, par le système informatique. Qu'il s'agisse de la collecte d'informations (surveillance de l'activité des serveurs Unix, des performances des éléments actifs du réseau, des liaisons et des flux d'informations circulant sur le Wan), de leur distribution (répercussion des alarmes par la messagerie, élaboration de tableaux de bord) et de l'automatisme des actions répétitives, le bilan technique est positif ».

6.4 La plate-forme OpenView de Hewlett-Pakcard

OpenView prend une place importante au palmarès des systèmes d'administration avec un engagement ancien et ferme pour le respect des standards : il a été choisi par l'OMG comme un composant du noyau DME. Il profite par ailleurs du support d'un grand nombre de sociétés tierces : IBM et Bull l'utilisent comme technologie de base pour leur propre système d'administration. OpenView a été porté sur les environnements de plusieurs constructeurs et particulièrement sur les stations de travail Sun. Il existe deux versions d'OpenView : celle appelée *Campus* limitée à la gestion de cinq cents nœuds et celle nommée *Enterprise* capable de gérer deux mille nœuds.

OpenView décompose le problème d'administration en trois composants de base :

- les applications administratives : elles dirigent les agents en leur demandant des informations ou des modifications à effectuer sur les objets. Les résultats sont proposés aux administrateurs grâce à l'interface graphique ;
- l'interface utilisateur graphique basée sur OSF/Motif avec le gestionnaire d'écran X/Windows. C'est par l'intermédiaire de l'API OVW (particulière à la plate-forme) que les applications administratives accèdent aux services graphiques d'OpenView ;
- le NMS (*Network Management Service*) : c'est le cœur du système qui lie l'ensemble des composants. Il se compose de plusieurs entités dont :
  - DCI (*Distributed Communication Service*),
  - DMS (*Data Management Service*).

Le DCI est le service de communication de la plate-forme. Disponible au travers de l'API XMP, il est basé sur les services CMIS. Toutes les entités de la plate-forme dialoguent à l'aide de ce service, excepté l'interface opérateur. Les protocoles SNMP et CMOT sont également proposés. Le sous-composant, nommé PostMaster, permet de localiser les instances d'objets et de cacher les détails d'exécution des services.

L'EMS (*Event Management Service*), composant essentiel du DCI, assure le routage des événements dans le réseau ainsi que leur stockage dans les journaux (*Events Logs*). Les événements sont les notifications (alarmes) émises de la part des agents à destination des applications administratives.

Le gestionnaire d'objets (ORS :*Object Registration Services*) est aussi un composant important du DCI. Il est chargé de gérer les instances d'objets représentant les ressources administrées. Il exécute les requêtes que lui adressent les applications administratives à destination des instances d'objets qu'il gère. Il relaie à destination

des applications administratives les événements émis par les agents ; il s'appuie évidemment sur le gestionnaire d'événements.

Le gestionnaire de données DMS fournit un accès vers une base de données Ingrès par l'intermédiaire de l'API standardisée SQL. La base de données contient la définition et la description des objets. DMS offre aussi aux applications administratives et au gestionnaire d'objets, un service de gestion d'objets présentant une interface de manipulation de données CMIS ; c'est le serveur Metadata qui gère ce service. L'accès aux données est ainsi transparent et indépendant de la localisation physique des informations et du SGBD utilisé.

La Figure 17 illustre l'architecture de la plate-forme OpenView.

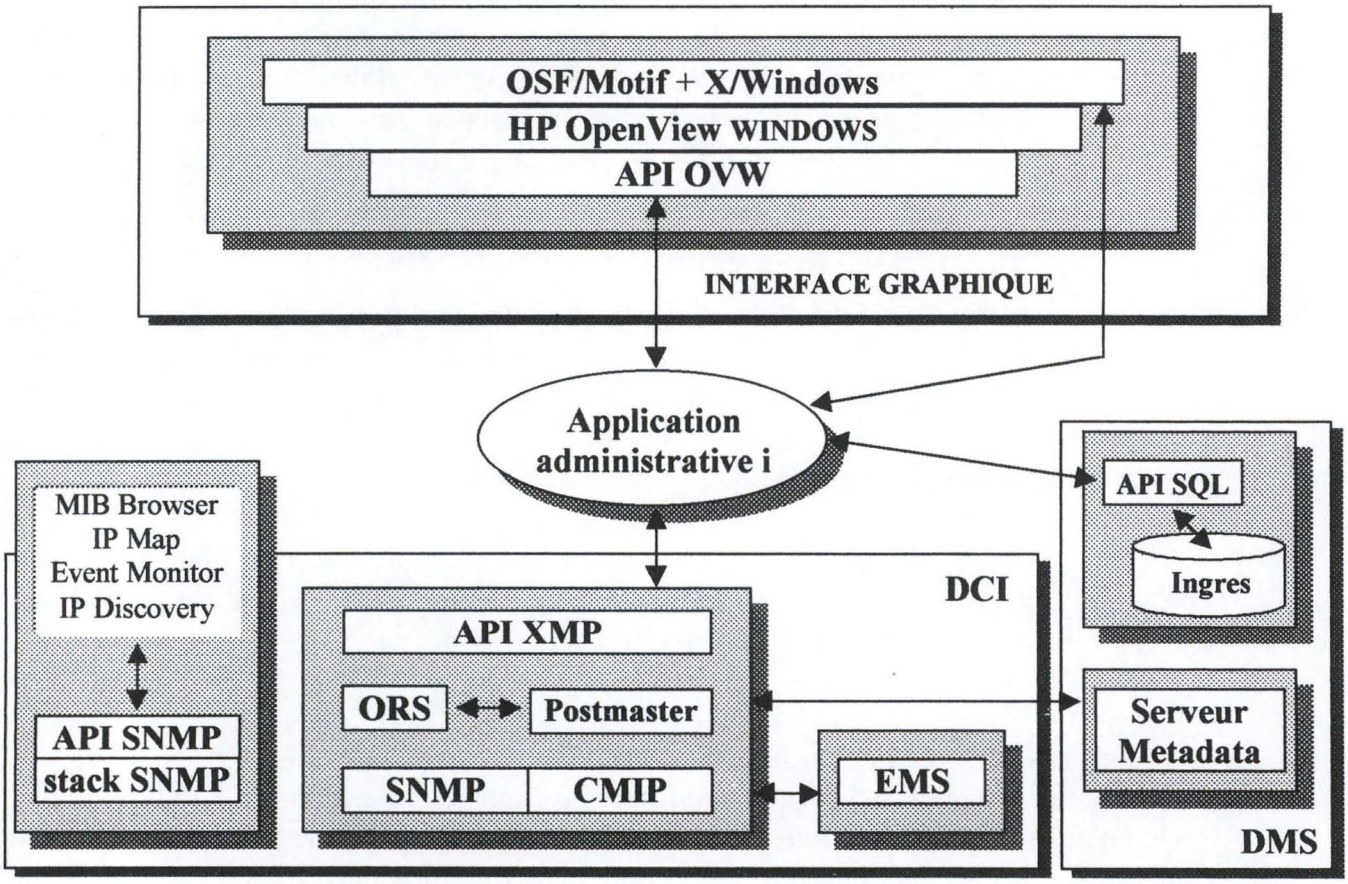


Figure 17 : Architecture d'OpenView

Voici quelques exemples de caractéristiques offertes par la plate-forme :

- **Carte réseaux**

  Les réseaux sont présentés comme une collection d'icônes (représentant les équipements) et de connexions. Les cartes sont structurées de manière

hiérarchique, ce qui permet une navigation plus aisée ; on peut ainsi acquérir des détails supplémentaires sur une partie de la carte (en cliquant sur un icône).

- **Recherche automatique de la topologie du réseau** (*Discovery*)

  La recherche automatique permet de récolter des informations du réseau sur lequel celle-ci est lancée. Les équipements, routeurs et liaisons sont les premiers éléments convoités par la recherche. Le résultat est un carte.

- **Sécurité de la console**

  Trois niveaux de sécurité sont proposés afin de permettre des niveaux d'accès différents. L'administrateur peut ainsi se connecter à la plate-forme avec le niveau de privilège le plus bas pour se mettre à l'abri des erreurs (par exemple, des modifications involontaires). Il peut aussi se déconnecter de la plate-forme, interdisant tout accès non autorisé, et permettre aux applications de continuer à surveiller le réseau.

- **Contrôle continu**

  Par la technique de *polling*, la plate-forme interroge périodiquement les périphériques et indique sur une carte et dans les journaux d'événements si ceux-ci sont opérationnels.

- **Indication d'urgence**

  Si un événement critique survient sur le réseau, la plate-forme est capable de prévenir immédiatement l'administrateur par l'intermédiaire d'un équipement spécifique (terminal, GSM, etc.)

En guise de conclusion, nous pouvons dire que la plate-forme HP OpenView se situe comme ISM, parmi les cinq meilleures du marché. Au début de l'année 1996, Hewlett-Packard annonçait une version d'OpenView pour le système d'exploitation Windows NT. Cette version était opérationnelle un an plus tard. Elle est spécialement conçue pour de petites et moyennes entreprises (PME). Le fait que HP porte sa plate-forme sur d'autres systèmes d'exploitation est la preuve que le marché de l'administration des réseaux est en pleine expansion.

6.5 Les agents intelligents

Comme nous l'avons vu, l'approche courante de la supervision de réseau comprend une plate-forme d'administration communiquant avec des agents. La plate-forme centrale est responsable de tous les calculs, de toutes les prises de décision, et ce sur un très grand nombre de variables et de conditions. A l'opposé, les agents sont des entités dénuées d'intelligence, transmettant seulement des variables de la MIB lorsque le gestionnaire les demande ou lors d'un événement particulier. Cette architecture présente de nombreuses faiblesses pour une gestion efficace de bon nombre de réseaux actuels. La gestion moderne demande une optimisation de l'utilisation des ressources ainsi qu'un traitement approprié et rapide des divers événements affectant le fonctionnement du réseau. La délégation et la distribution de l'intelligence peuvent permettre d'améliorer l'architecture actuelle. Le but est donc ici d'apporter de nouvelles fonctionnalités aux agents, de leur permettre de prendre des décisions face à certains événements, de les rendre plus autonomes et de réduire la tâche de la plate-forme centrale, comme l'illustre la Figure 18.

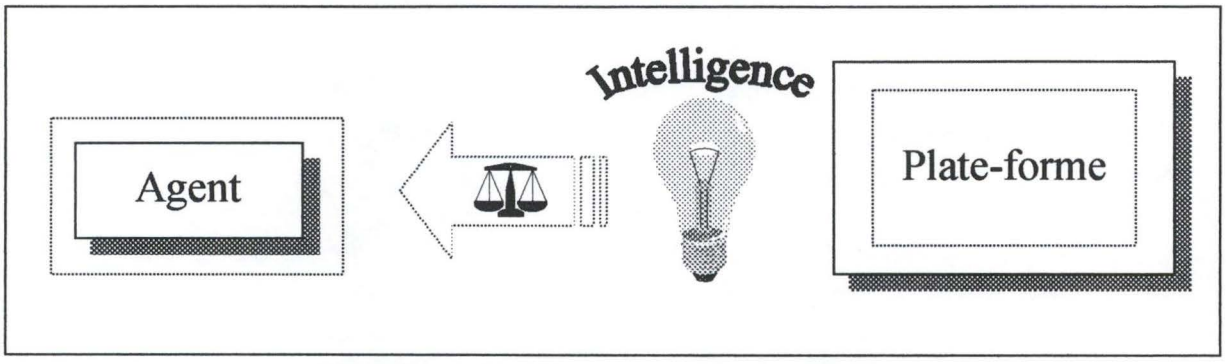


Figure 18 : Equilibrage de l'intelligence

L'utilisation de la délégation permet aux agents d'acquérir une certaine autonomie. Par exemple, lorsque les liaisons sont coupées avec la plate-forme, un agent peut activer des fonctions qui fournissent des instructions de gestion prédéfinies pour un fonctionnement en mode autonome. Par exemple, un agent délégué responsable d'une partie du réseau (un gestionnaire local en quelque sorte) peut continuer à accumuler des informations et à prendre des décisions, même s'il est isolé (à cause d'une panne) de la plate-forme centrale.

La délégation permet aussi de gérer plus efficacement un réseau lorsque les performances évoluent en fonction du temps. La possibilité de transférer des fonctions aux agents et d'accéder à celles-ci pendant les périodes critiques pour le réseau (saturation, défaillance, etc.) réduit la bande passante qui serait consommée par une approche centralisée. Lorsque le réseau est chargé, le trafic additionnel généré par les requêtes de supervision peut devenir significatif. De plus, en période de fortes charges

ou de saturation, sans un traitement approprié et rapide, les problèmes tendent à empirer et les ressources du réseau à devenir de moins en moins disponibles.

Pour une meilleure distribution de l'intelligence, il faut que la fonction de gestion ait les caractéristiques suivantes :

- *Autonomie locale* : la plupart des informations doivent être disponibles localement, et la gestion ne doit nécessiter qu'une bande passante très faible en dehors du domaine local ;
- *Stabilisation locale* : si la source du problème est locale, le domaine doit être capable de prendre des décisions pour corriger les problème localement ;
- *Dégradation progressive* : la fonction de gestion doit être efficace et les services réseau doivent continuer, même avec des performances amoindries, lorsque les conditions empirent. Cela nécessite une architecture distribuée, avec un faible besoin de ressources éloignées ;
- *Anticipation des problèmes* : la surveillance distribuée permet aux domaines locaux de prévoir les conditions défavorables avant qu'elles ne surviennent. Les mesures correctives peuvent être prises localement ou nécessiter une interaction entre domaines.

Une technique pour la surveillance des surcharges du réseau est la corrélation des variables de la MIB reflétant les problèmes locaux, tels que les retransmissions, les pertes ou encore les *timeout*[20].

Certains problèmes ne peuvent être résolus qu'à partir de la vue globale du réseau ; autrement dit, la délégation des fonctions de gestion possède des limites. Du « tout centralisé » au « tout distribué », l'administrateur devra trouver les meilleurs compromis afin que la délégation de l'intelligence améliore la consommation de la bande passante et diminue les temps de réaction face aux dysfonctionnements, sans pour autant dégrader la vue globale du système.

6.6 Conclusion

Maîtriser la complexité et l'évolution des réseaux constitués de nombreux équipements hétérogènes passe généralement par l'acquisition d'une plate-forme d'administration. Elle donne les moyens aux administrateurs de configurer, surveiller et surtout de faire évoluer le système informatique, pour que celui-ci réponde au mieux aux besoins spécifiques des clients.

---

[20] Le *timeout* permet la génération d'un événement après l'écoulement d'un laps de temps. Par exemple, les entités protocolaires de transport utilisent des *timeout* pour la retransmission d'un paquet de données.

Chapitre 7 : La distribution de logiciels

7.1 Introduction

Ce chapitre a un double objectif : premièrement, présenter un exemple d'application développée à l'aide des outils génériques de la plate-forme ISM et deuxièmement, présenter l'application que j'ai développée durant mon stage de fin d'études.

Nous commencerons ce chapitre en expliquant les principes de base concernant la distribution de logiciels. Nous présenterons ultérieurement le produit SDPC[21] de la société Bull développé dans le cadre d'ISM. Lors de l'analyse de l'architecture de l'application, nous nous rendrons compte d'une faiblesse essentielle : un manque d'information. Nous proposerons une solution à cette défaillance et comment celle-ci a été implementée. La conclusion clôturera ce chapitre.

7.2 Les principes de base

Lors des chapitres précédents, nous avons fait remarquer qu'une propriété importante de l'administration était la centralisation du contrôle. La distribution de logiciels[22] (*software distribution*) s'intègre parfaitement dans le domaine de l'administration puisqu'elle permet de centraliser l'installation de logiciels sur de multiples machines connectées au réseau. L'administrateur utilise une application de la plate-forme (que nous appellerons *application interface*) pour spécifier quels sont les logiciels à installer et quelles sont les machines concernées (voir la Figure 19 de la page suivante). Il peut ensuite visualiser en temps réel l'état d'avancement des processus d'installation.

Quel est l'avantage d'une distribution et d'une installation de logiciels centralisées ? Nous pouvons répondre à cette question en faisant référence à un courrier électronique reçu récemment par l'ensemble des étudiants de notre institut d'informatique[23].

✉ « Je cherche pour le mois de septembre un(e) étudiant(e) de dernière année intéressé(e) pour l'installation sur une **trentaine de PC** de la configuration suivante : Win95 OSR2, Office Pro 97, client Novell et divers utilitaires. Ce job rémunéré est offert à Namur et est estimé à environ **une dizaine de jours** ... »

[21] SDPC (*Software Distribution PC*).
[22] Le terme « distribution de logiciels » regroupe les étapes de distribution, d'installation et de suivi.
[23] L'auteur de ce courrier est le responsable des ressources informatiques, M. Cotet.

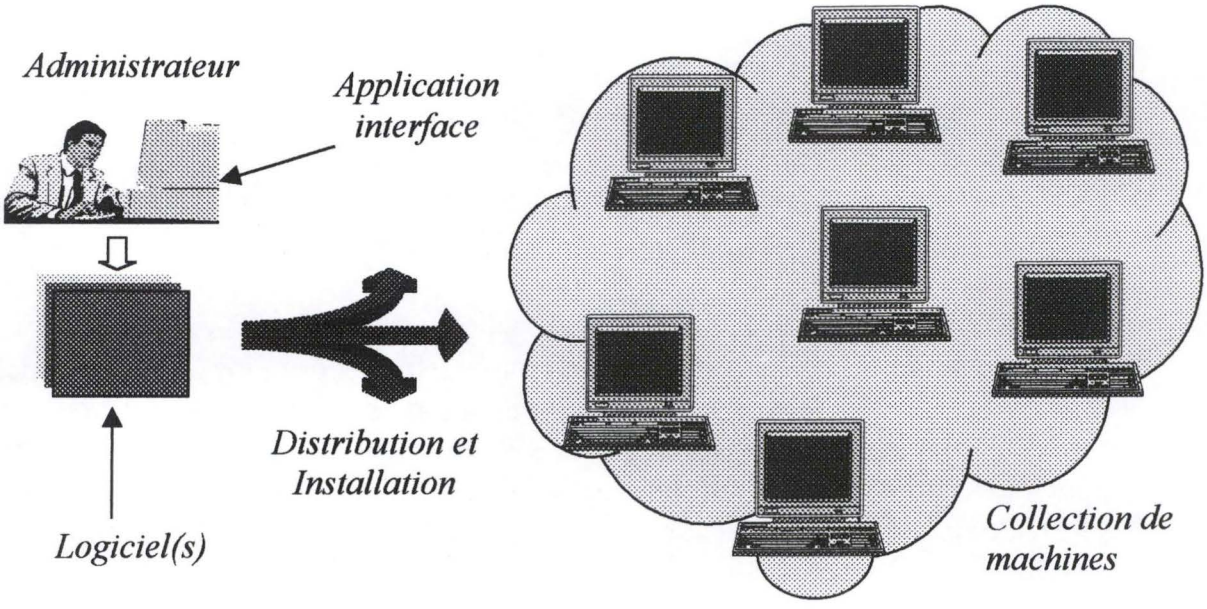


Figure 19 : Vue schématique de la distribution de logiciels

Si une dizaine de jours sont prévus pour trente PC, combien faudrait-il de jours ou de mois aux personnes responsables de plusieurs milliers de machines ?

Nous savons que le monde de l'informatique est perpétuellement en évolution ; de nouvelles versions des logiciels sortent pratiquement chaque année. Lorsque le nombre de machines concernées par ces évolutions dépassent un certain seuil, les responsables du parc informatique doivent se demander si l'achat d'un logiciel de distribution n'est pas plus rentable que des installations manuelles. La réponse à cette question dépend principalement de la stratégie informatique de la société, du nombre de machines qu'elle possède et du nombre de logiciels installés par unité de temps (par exemple, deux logiciels par mois).

7.3 SDPC : La solution de Bull

La société Bull, dans le cadre d'ISM, a développé deux applications distinctes pour la distribution et l'installation automatiques de logiciels :

- ISM SD (*ISM Software Distribution*) pour les serveurs UNIX ; et
- ISM SDPC (*ISM Software Distribution for PC*) pour les PC.

Nous limiterons notre étude à SDPC puisque c'est sur cette application que j'ai réalisé mon stage de fin d'études. Notons seulement que les caractéristiques et les concepts fondamentaux de SDPC sont valables pour SD.

SDPC est le composant d'ISM PC OperationMaster[24] responsable de la distribution et l'installation automatiques de logiciels sur les PC à travers un réseau de communication. Ses caractéristiques principales sont les suivantes.

- SDPC est une solution simple pour installer des logiciels sur des PC indépendamment de leur protocole de communication et de leur système d'exploitation. Autrement dit, SDPC s'exécute dans un environnement hétérogène.
- SDPC permet de préparer et d'exécuter la distribution d'une manière centralisée.
- Intégré dans ISM OpenMaster, SDPC tire un maximum d'avantages des technologies et des services disponibles : surveillance des réseaux, génération d'alarmes, etc.
- L'administrateur ou l'opérateur peuvent utiliser ISM Monitor et ISM Alarm pour surveiller tous les éléments qui interviennent dans la distribution tels que les serveurs et les PC cibles ainsi que l'ensemble des éléments qui participent aux transferts des données : routeurs, ponts, etc.
- SDPC simplifie considérablement le travail de l'opérateur. Celui-ci spécifie simplement quels sont les produits à installer et les PC concernés. Le serveur où sont localisés les fichiers à transférer est automatiquement déterminé par les algorithmes de l'application.

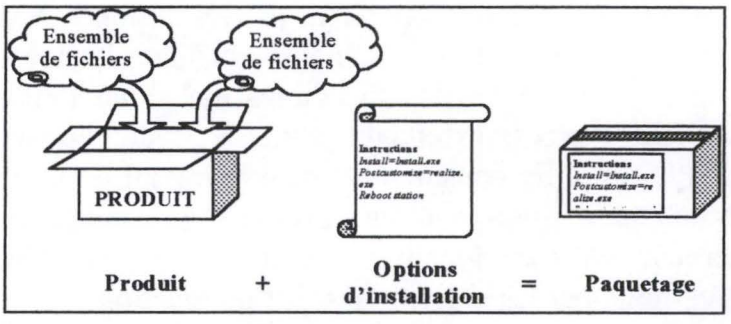
[24] Les différents modules d'ISM OpenMaster, dont PC OperationMaster ont été expliqués à la page 63.

7.3.1 Les différentes étapes de la distribution

Les éléments principaux de SDPC sont constitués de l'application interface, des serveurs d'administration (*admin servers*) et de dépôt (*repository servers*) ainsi que des stations de travail (*workstations*). L'application interface s'exécute sur le serveur ISM (celui où se trouve la plate-forme ISM) ; cette application possède une interface graphique conviviale. Les rôles des deux serveurs seront étudiés ultérieurement. Les stations de travail désignent les machines cibles de la distribution.

Les étapes principales de la distribution, avec SDPC, sont les suivantes.

- La première étape concerne le stockage des logiciels à distribuer sur le serveur ISM. Par exemple, l'administrateur devra copier les fichiers des logiciels sur le serveur à partir des disquettes ou du CD-ROM. Il pourra aussi spécifier les fichiers concernés par la distribution. Cette sélection est utilisée, par exemple, pour installer un logiciel sans ses extensions (documentation complémentaire, bibliothèque d'images, etc.)
- La deuxième étape permet de définir les paquetages (*packages*). Ceux-ci se composent d'un ensemble de sous-produits (les logiciels) et de composants (les drivers par exemple). Les paquetages permettent non seulement d'installer plusieurs produits en une seule opération mais aussi de remplacer, lors d'un changement de version, l'ensemble des produits affectés par la modification. Les paquetages peuvent être dépendants ou autonomes. Lors de la création d'un paquetage, l'administrateur peut spécifier des options d'installations



- La troisième étape permet de transférer les fichiers des logiciels (préparés lors de la première étape) sur des serveurs de dépôts.
- La quatrième étape concerne la définition des travaux (*jobs*). Ceux-ci permettent de sélectionner les paquetages et spécifier les PC sur lesquelles ceux-ci seront installés.
- La cinquième étape est l'envoi des instructions par l'application SDPC aux serveurs d'administration appropriés.
- Lors de la sixième et dernière étape, les PC concernés reçoivent, de leur serveur d'administration, leurs instructions ; accèdent aux serveurs de dépôts spécifiés dans les instructions ; et effectuent les différentes installations.

Les trois premières étapes ne sont effectuées que lors de l'ajout d'un nouveau logiciel au système de distribution. Elles sont réalisées par l'administrateur car elles nécessitent des connaissances particulières des logiciels. Par contre, la quatrième étape est une activité journalière et ne nécessite pas de connaissances spécifiques. Elle est effectuée par l'opérateur via l'interface graphique de SDPC. Les deux dernières étapes sont complètement automatisées par l'application SDPC. Elles ne nécessitent aucune intervention de la part de l'opérateur. Néanmoins, certaines installations requièrent des interactions avec l'utilisateur final du PC.

Les informations sur le déroulement des installations sont stockées sur les serveurs d'administrations. Les informations sont remontées jusqu'à la plate-forme par l'intermédiaire d'alarmes et sont disponibles pour l'administrateur via l'application interface de SDPC. La "granularité" (dans le sens gradation) des informations dépend de la configuration des serveurs d'administration.

7.3.2 Les serveurs de l'application

Le serveur ISM, appelé aussi serveur principal, est celui où s'exécute la plate-forme et, entre autre l'application interface de SDPC. Ce serveur ne communique pas directement avec les stations de travail. Celles-ci n'échangent des informations qu'avec les serveurs SDPC (généralement des serveurs Windows NT ou Netware), comme l'illustre la Figure 20 ci-dessous.

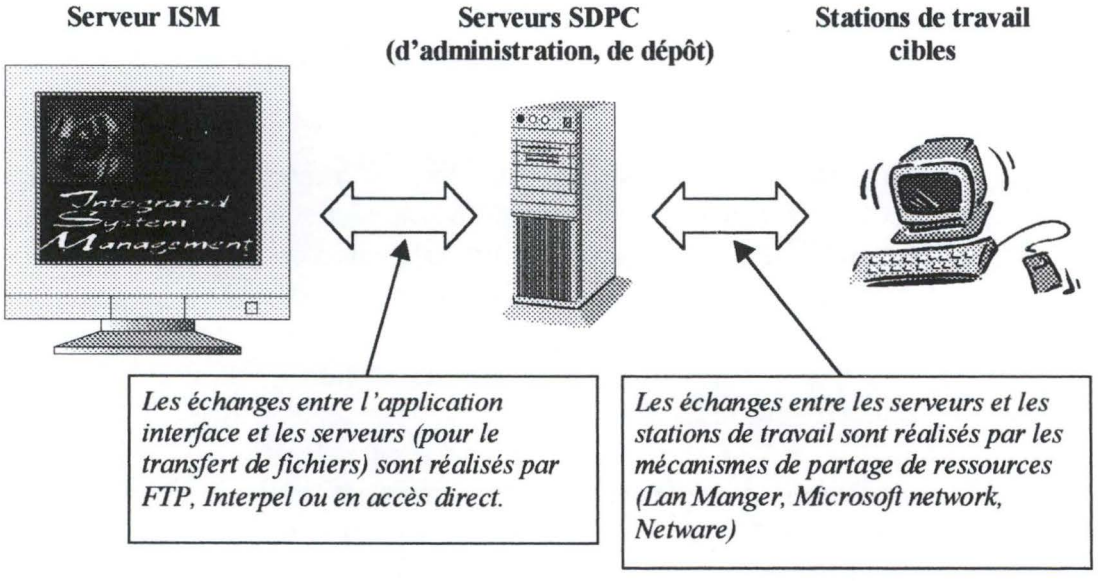


Figure 20 : Architecture de communication de SDPC

La caractéristique principale des serveurs SDPC est qu'ils sont accessibles à la fois par le serveur ISM et par les stations de travail. Les ordres d'installation sont

envoyés par l'application interface aux serveurs SDPC où les stations peuvent venir les retirer. Les ordres spécifient les instructions (installer un logiciel, désinstaller un logiciel, etc.) qui doivent être effectuées sur les stations de travail. C'est une application cliente, s'exécutant sur les stations (*sdclient*), qui exécute les instructions contenues dans les ordres.

La ressource particulière où sont stockés les ordres est appelée la **ressource d'administration** (*admin ressource*). Un serveur SDPC qui possède une telle ressource est dit **serveur d'administration** (*admin server*). Dans cette ressource, chaque station de travail possède son propre répertoire dans lequel elle peut retirer ses ordres d'installations. Ce principe est similaire à celui des boîtes aux lettres. Une propriété importante est qu'une station de travail, pour faire partie du système de distribution, doit être rattachée à un et un seul serveur d'administration.

Lorsqu'un ordre nécessite l'installation d'un logiciel, il contient des informations sur l'endroit où sont stockés les fichiers d'installation. L'endroit de stockage est un serveur SDPC appelé **serveur de dépôt** (*repository server* ou *deposit server*). Chaque serveur de dépôt fournit une ou plusieurs **ressources de dépôt** (*deposit ressources*).

Un serveur SDPC peut être à la fois un serveur d'administration et un serveur de dépôt. Notons aussi qu'un PC peut être une station de travail et un serveur SDPC.

7.3.3 Un scénario d'installation

Avant et pendant l'installation d'un logiciel, l'opérateur a la possibilité, par l'intermédiaire de l'application interface :

- de consulter les temps estimés pour les différentes installations et de suivre, en temps réel, l'état d'avancement de celles-ci ;
- de vérifier l'état d'une station de travail et de consulter quel est son système d'exploitation, quelle est sa capacité de stockage, etc.

La Figure 21 de la page suivante illustre un scénario complet d'installation. Les différentes opérations sont les suivantes :

[1] L'administrateur copie les fichiers d'installation du logiciel, appelons-le A, sur le serveur ISM et les transfère sur les serveurs de dépôts (le transfert est réalisé par l'application SDPC soit via *ftp* soit via *Interpel*). Ensuite, il crée le paquetage nécessaire (comprenant le logiciel A) et spécifie éventuellement les options d'installation.

[2] L'opérateur crée un ordre spécifiant que le logiciel A doit être installé sur notre station de travail. L'ordre est ensuite automatiquement envoyé au serveur d'administration de notre station.

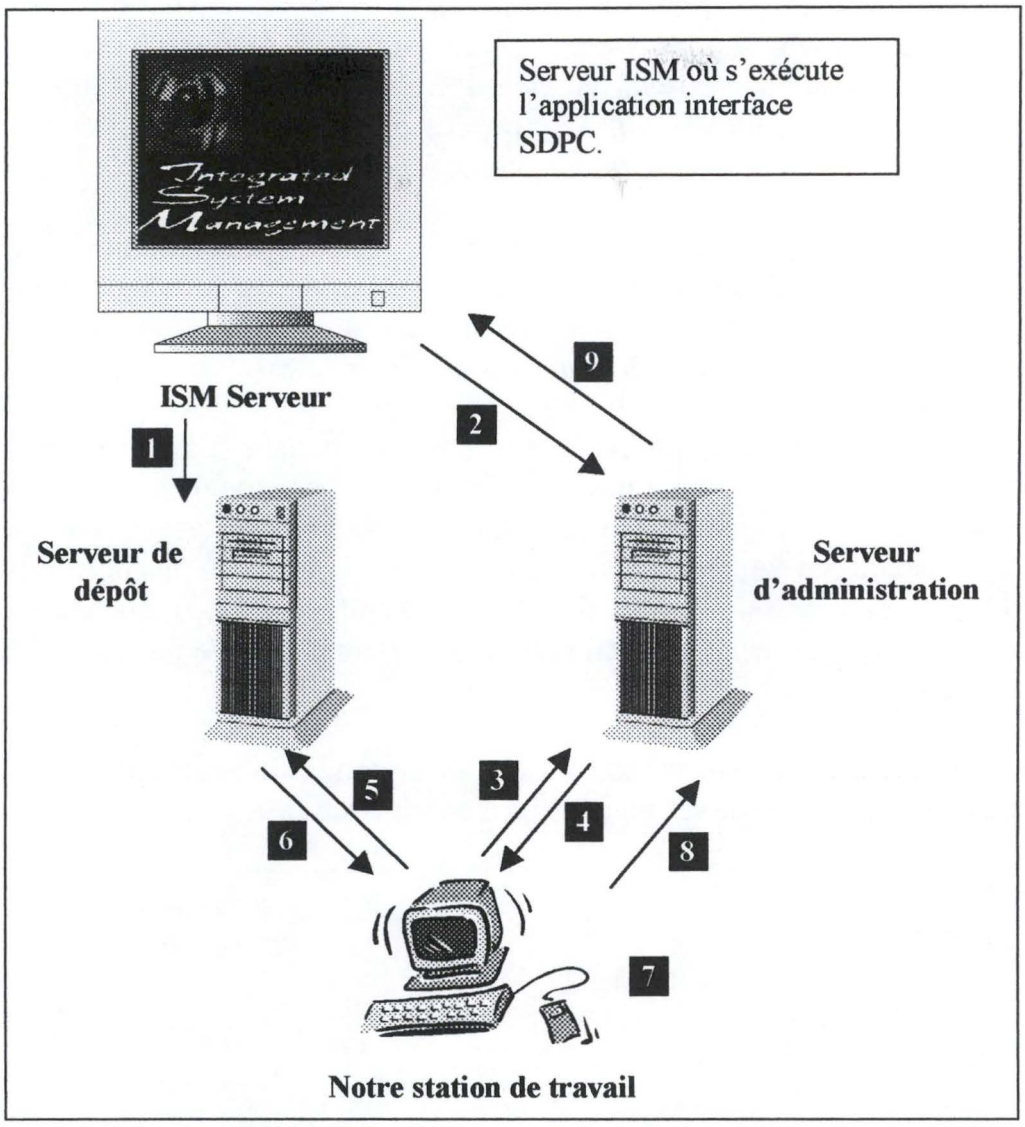


Figure 21 : Scénario d'une installation

[3] Lors du démarrage ou après un certain délai, le processus *sdclient* de notre station contacte son serveur d'administration et demande les ordres qui lui sont destinés.

[4] Le serveur d'administration envoie l'ensemble des ordres spécifiques à notre station (ceux-ci comprennent au moins, dans notre cas, l'ordre d'installation du logiciel A).

[5] Dans l'ordre se trouve l'indication du serveur de dépôt le plus proche contenant les fichiers d'installation du logiciel A. Notre processus *sdclient* contacte ce serveur et demande l'accès à ces fichiers.

[6] Le serveur de dépôt partage le répertoire nécessaire afin que notre station puisse y accéder par le réseau (le partage des ressources se fait via des protocoles tels que Lan Manager, Netware ou Microsoft network).

[7] L'installation du logiciel A sur notre station commence.

[8] Les erreurs et l'état d'avancement de l'installation (début, fin de l'installation) sont remontés de la station vers le serveur d'administration.

[9] Le serveur d'administration, par rapport à sa configuration, remonte les informations (erreurs et état d'avancement) vers la plate-forme via des alarmes. Celles-ci sont interprétées par l'application interface et présentées sous forme textuelle ou graphique à l'opérateur.

SDPC garde un historique de toutes les sessions d'installation déjà effectuées. L'opérateur a ainsi la possibilité de consulter les anciennes installations (paquetages et machines cibles), de consulter l'ensemble des propriétés d'un paquetage (fichier, options, etc.) et, s'il le désire, de réutiliser un d'eux.

SDPC gère aussi la notion de prérequis permettant d'éviter les problèmes lors des installations. Ceux-ci sont utilisés, par exemple, pour vérifier que la configuration des machines est compatible avec l'installation du nouveau paquetage ; on peut ainsi éviter d'installer un logiciel spécifique à Windows NT sur un machine Windows 3.11. SDPC permet aussi de lancer des *scripts* avant (*pre-load*) ou après (*post-load*) l'installation. Ces *scripts* sont similaires aux fichiers *batch* du DOS (fichiers avec l'extension .BAT).

La Figure 22 ci-dessous illustre la structure hiérarchique de la ressource d'administration présente sur un serveur d'administration.

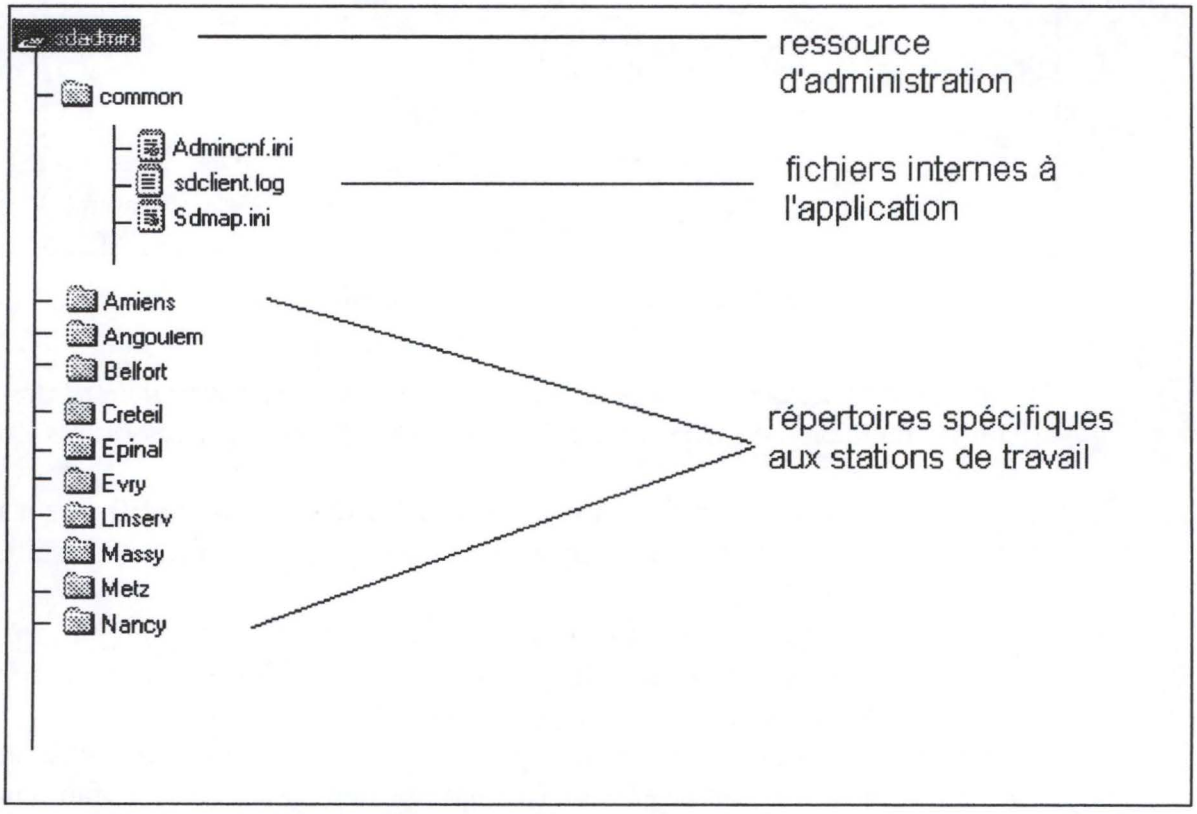


Figure 22 : Structure hiérarchique de la ressource d'administration

7.3.4 L'architecture de l'application

SDPC est composé de plusieurs applications s'exécutant sur le serveur ISM. Utilisant les applications appropriées de SDPC, l'administrateur déclare les différents serveurs, les stations de travail, prépare les paquetages, transfère les logiciels sur les serveurs de dépôt, etc.

Une fois que le travail de l'administrateur est terminé, l'opérateur peut définir des travaux (des ordres d'installations) en sélectionnant les paquetages à distribuer et les stations de travail concernées.

Les applications de SDPC partagent une base de données commune qui contient les informations sur les différentes ressources (serveur, paquetage, station de travail, etc.) ainsi que sur les travaux. Cette base de donnée est gérée par le SGBD ORACLE. La Figure 23 représente l'architecture de SDPC.

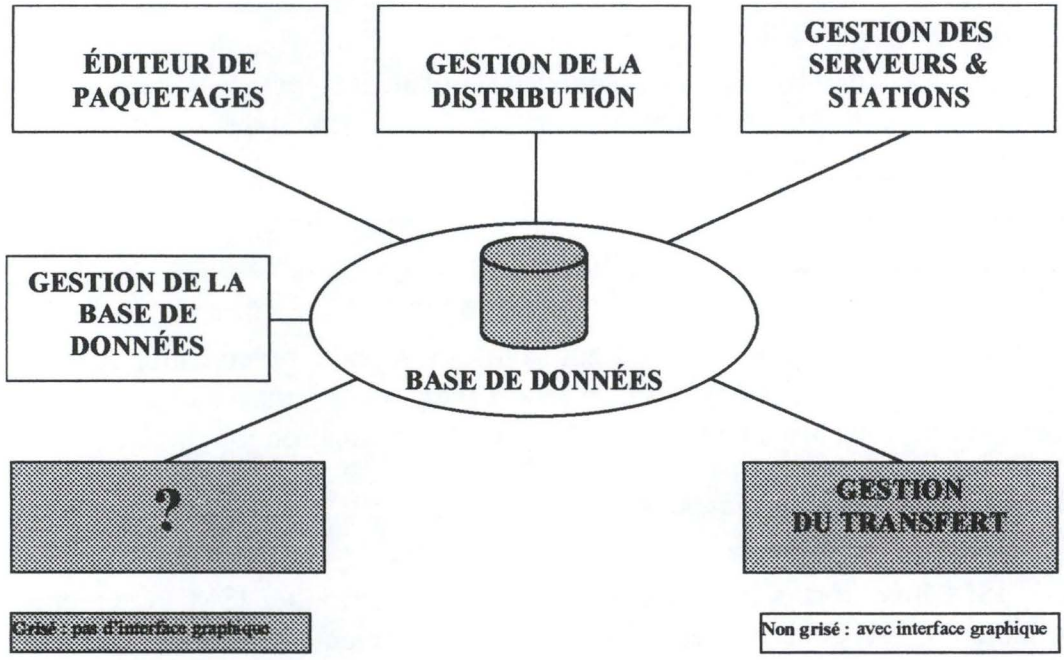


Figure 23 : L'architecture de SDPC

Le tableau ci-dessous reprend pour chaque application ses fonctions principales ainsi que la personne concernée.

Application SDPC	Principales fonctions	Personne concernée
Gestion de la BD	Créer et supprimer des BD	Administrateur
Editeur de paquetages	Définir les produits et les paquetages	Administrateur
Gestion des serveurs & stations	Définir et configurer les stations, les serveurs et leurs ressources.	Administrateur

Gestion de la distribution	Définir les travaux et consulter leurs suivis.	Opérateur
Gestion du transfert	Exécuter l'ensemble des transferts (fichiers et travaux)	Administrateur (activée par une horloge[25])

Malgré ses puissantes possibilités, SDPC présentait la faiblesse d'être incapable d'offrir à l'administrateur ou à l'opérateur des informations détaillées sur les stations de travail ou les serveurs impliqués dans la distribution. Pourtant, lorsque l'opérateur décide de distribuer un logiciel, il serait intéressant de lui procurer des informations sur les PC potentiels à la distribution. En effet, une information sur le type de l'ordinateur (Intel DX33 par exemple) peut amener l'opérateur à ne pas installer une version d'un logiciel qui serait trop lente sur ce type de machine.

Cette défaillance était amplifiée par le fait qu'une application indépendante de SDPC, l'application inventaire (*ISM Inventory*), disposait de ces informations. Les concepteurs de SDPC ont donc décidé de créer une nouvelle application dont le rôle serait de rapatrier ces informations dans la base de données de SDPC. Dans la Figure 23 (page précédente), celle-ci, appelée **consolider**, est représentée par le cadre contenant le point d'interrogation. Le sujet de mon stage a été de concevoir et d'implémenter cette application.

7.3.5 Le consolider

Comme nous l'avons souligné dans la section précédente, le consolider est l'application interface entre SDPC et ISM Inventory. Avant d'examiner plus en détails le consolider, nous allons analyser brièvement l'application inventaire.

7.3.5.1 L'application inventaire

ISM Inventory s'intègre dans le module fonctionnel ISM PC OperationMaster. Son objectif est de créer et de tenir à jour l'inventaire des différentes machines présentes sur le réseau. L'inventaire est constitué essentiellement des informations matérielles (CPU, mémoire, ports de communication, etc.) et logicielles (les applications installées sur la machine).

Dans SDPC, chaque station de travail possède un serveur d'administration, dépositaire des ordres et des informations de suivi. ISM Inventory est basé sur le même concept : chaque machine est rattachée à un **serveur d'inventaire** dépositaire des informations inventoriées (voir la Figure 24 de la page suivante).

[25] L'utilitaire **crontab** disponible sur un système d'exploitation Unix permet d'activer un processus à une date fixe ou à intervalles réguliers (par exemple toutes les heures).

Sur chaque machine à inventorier (station de travail ou serveur) se trouve un processus responsable de collecter les informations et de les transférer au serveur d'inventaire approprié ; le transfert s'effectue au moyen d'un fichier plat[26]. Le serveur d'inventaire extrait les informations de ce fichier et les stocke dans une MIB commune à toutes les machines. Les informations de la MIB peuvent être ensuite consultées par l'intermédiaire d'un agent s'exécutant sur le serveur d'inventaire.

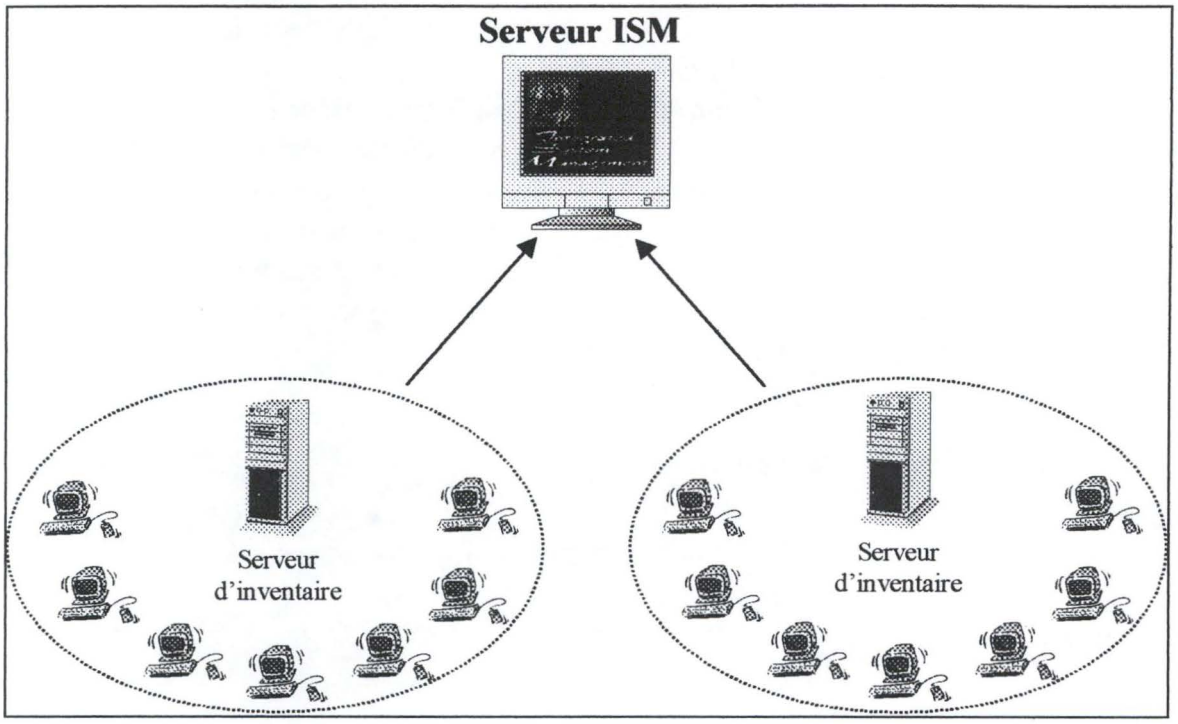


Figure 24 : Architecture simplifiée de l'inventaire

Grâce à l'application interface d'ISM Inventory (qui s'exécute sur le serveur ISM), l'opérateur peut consulter les inventaires des différentes machines.

7.3.5.2 Les spécifications du consolider

Les spécifications du consolider étaient les suivantes : la base de données commune de SDPC contient une table comportant l'ensemble des machines appartenant au système de distribution ; le consolider doit, à partir des informations contenues dans cette table, contacter le serveur d'inventaire de chaque machine, rapatrier les informations inventoriées et les insérer dans la base de données. Il doit aussi créer un rapport résumant le déroulement des opérations : « quelles est le nombre et la liste des machines inventoriées », « quelles sont les machines non inventoriées

[26] Un fichier plat est un fichier ne possédant aucune structure particulière, contrairement à une base de donnée. Les informations sont stockées les unes à la suite des autres. Le découpage de l'information en champs est réalisé à l'aide de caractères séparateurs (retour de chariot, virgule, etc.).

ainsi que la cause des échecs », ainsi que des indications sur les performances (par exemple, *n* secondes pour l'inventaire de la machine B).

Les contraintes qualitatives et quantitatives du consolider étaient les suivantes :

- L'insertion de l'inventaire d'une machine dans la base de données devait être atomique : soit toutes les informations de la machine sont insérées (logiciels, matériels, etc.), soit aucune. Cette propriété permet de garantir la cohérence de la base de données.
- L'inventaire d'une machine doit être efficace et consommer le moins possible de bande passante (n'oublions pas que l'administration doit minimiser les impacts sur les performances du réseau) .
- L'exécution du consolider doit être transparente à l'opérateur. Cette contrainte a été solutionnée par un lancement automatique du consolider.
- Si le consolider ne parvient pas à inventorier une machine, il devra réessayer lors de sa prochaine exécution.

7.3.5.3 Le fonctionnement du consolider

La Figure 25 ci-dessous schématise le fonctionnement du consolider.

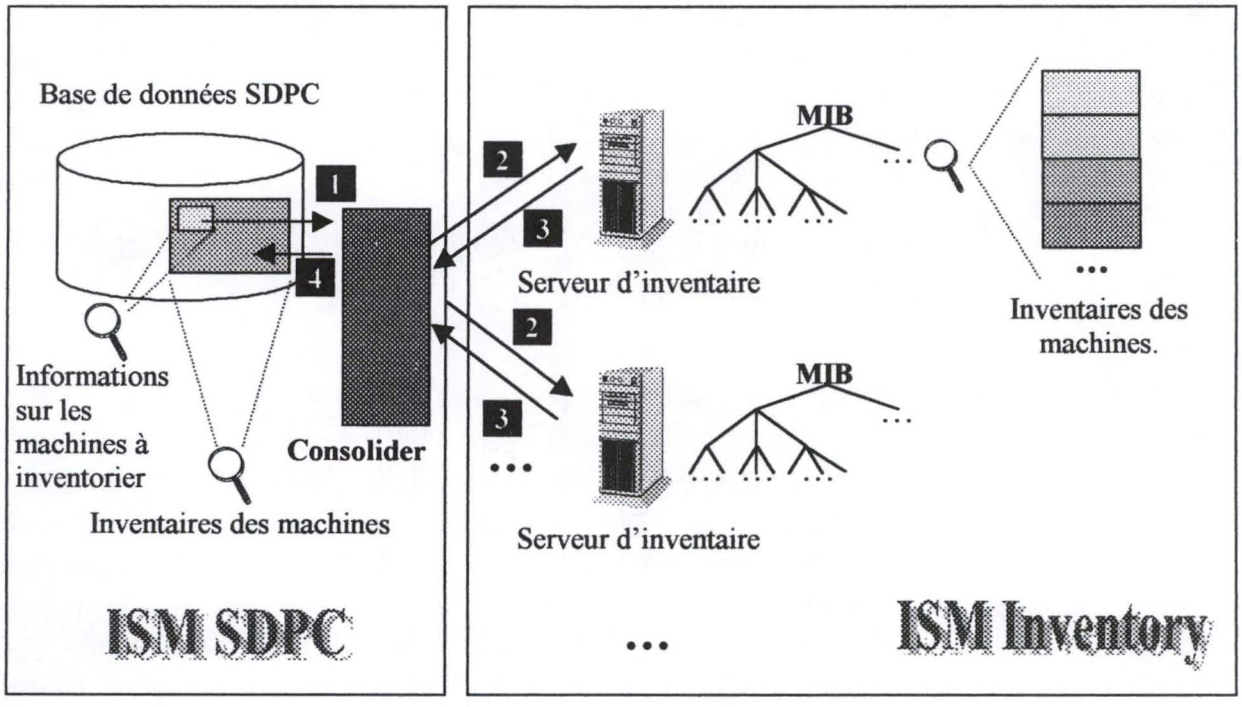


Figure 25 : Schéma du fonctionnement du consolider

Les principales opérations réalisées par le consolider sont les suivantes :

[1] Le consolider extrait de la base de données commune de SDPC les informations sur les stations à inventorier. Celles-ci sont constituées principalement de l'identifiant de la machine et celui de son serveur d'inventaire.
[2] Ensuite, les requêtes CMIS sont initialisées et lancées sur le réseau à destination des serveurs d'inventaire.
[3] Ceux-ci extraient de la MIB les informations requises et les retournent au consolider.
[4] Chaque fois que le consolider reçoit **toutes** les informations pour une machine, il les insère, au sein d'une même transaction, dans la base de données SDPC.

7.3.5.4 Les optimisations du consolider

Le caractère asynchrone de la plate-forme ISM permet d'émettre simultanément plusieurs requêtes CMIS. Le nombre de requêtes que le consolider peut lancer en parallèle dépend de deux contraintes antagonistes : d'une part le caractère rapide du consolider préconise l'envoi simultané de toutes les requêtes et d'autre part, la consommation de la bande passante la plus minimale possible prône pour l'exécution séquentielle des requêtes. Nous avons opté pour un compromis : exécuter un nombre fixe de requêtes simultanées, ce nombre étant "paramétrable" (voir Figure 26 ci-dessous).

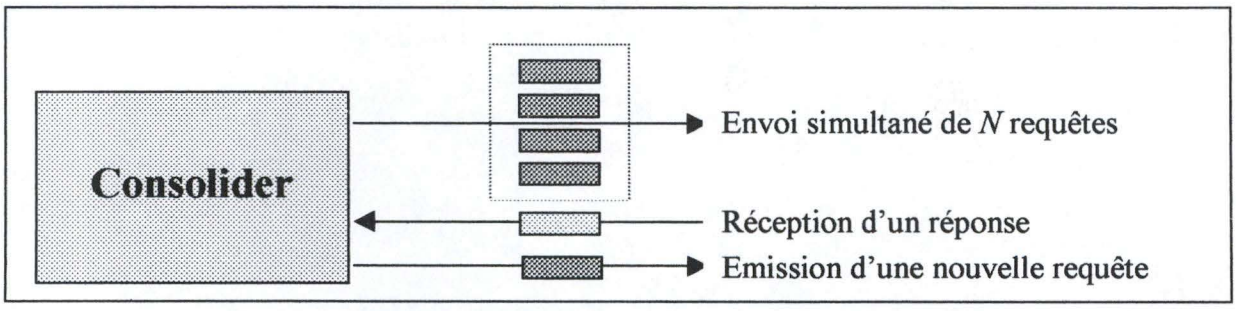


Figure 26 : Principe de l'émission des requêtes CMIS

Paradoxalement, le lancement de plusieurs requêtes peut dégrader les performances du consolider. En effet, si celles-ci sont émises vers le même serveur d'inventaire alors elles sont traitées séquentiellement par l'unique agent de ce serveur. Autrement dit, la première requête est traitée immédiatement et les autres sont stockées dans une file d'attente. Pour tirer un avantage du parallélisme, il est nécessaire d'envoyer les requêtes vers des serveurs d'inventaires différents.

Pour parvenir à ce résultat, le consolider, après avoir extrait de la base de données SDPC les informations sur les machines à inventorier, regroupe celles-ci en

fonction de leur serveur d'inventaire. Cela permet au consolider d'émettre uniquement une et une seule requête vers un même serveur d'inventaire.

Si un serveur ne répond pas, alors le consolider abandonne l'inventaire de toutes les machines rattachées à celui-ci. Cela permet de ne pas surcharger le réseau avec des requêtes "condamnées" d'avance. De nouvelles tentatives seront, de toute façon, effectuées lors d'une prochaine exécution du consolider.

D'autres optimisations ont été réalisées au sein du consolider mais celles-ci restent trop techniques pour être exposées dans ce mémoire.

7.3.5.5 Conclusion

Le développement d'une application sur la plate-forme d'administration ISM a été pour moi une expérience très séduisante. L'utilisation des différents outils génériques (base de données, requêtes CMIS, etc.) m'a permis de me pencher plus sur les problèmes conceptuels que sur les détails techniques. Le consolider a été totalement implémenté, testé et documenté avant la fin du stage. Les optimisations apportées ont permis d'augmenter la vitesse d'exécution d'un facteur vingt.

Le consolider, bien qu'il puisse paraître simple, est construit sur la base de plus de cinq mille lignes de code SML. La principale cause de cette envergure est qu'il doit être capable de faire face à un maximum d'erreurs ; n'oublions pas que cette application sera vendue à des clients. L'amélioration principale qu'il risque de subir dans l'avenir, est la présentation du rapport sous un forme HTLM. Cela permettra à l'administrateur de consulter celui-ci de n'importe où dans le monde.

Chapitre 8 : Conclusion générale

Ces dernières années, nous avons été témoins d'avancées technologiques essentielles ouvrant la voie aux systèmes ouverts et à la répartition. Nous avons assisté à la transition d'environnements de groupes de terminaux passifs vers une multiplicité de stations de travail variées, interconnectées et dotées de plus en plus d'autonomie. Le coût de cette migration a été principalement un accroissement de l'hétérogénéité des systèmes. Les responsables de ces derniers ont dû se poser de nouvelles questions :

- Comment assurer le fonctionnement de cet ensemble hétérogène ?
- Comment garantir le fonctionnement normal de chaque composant du système global conformément aux attentes des utilisateurs ?

L'administration des réseaux constitue l'un des points clés de ces intérrogations.

La première partie de ce mémoire a introduit la problématique de l'administration d'un réseau en mettant l'accent sur le *quoi* et le *pourquoi* avant de s'interroger sur le *comment* faire.

L'introduction, en plus de définir l'administration, a montré l'importante de sa centralisation. C'est à partir de la vue globale du système que l'administrateur peut comprendre les problèmes et déterminer les solutions les plus adéquates.

Dans le deuxième chapitre, les composants administrables et les fonctionnalités nous ont permis de cerner le domaine d'application de l'administration.

Ensuite, dans le troisième chapitre, nous avons dégagé les enjeux et les finalités de l'administration : assurer et maintenir les niveaux de qualité de service requis par les utilisateurs du réseau. Nous avons également présenté quatre critères génériques qui permettent d'évaluer la continuité du service (la disponibilité), sa fiabilité et son efficacité (délai et capacité). La négociation de la qualité de service des réseaux a été aussi abordée dans ce chapitre.

Le chapitre quatre met en évidence l'importance du système d'information pour atteindre les finalités souhaitées. Celui-ci représente et décrit l'état et le comportement des différents composants du réseau. Notre étude s'est limitée au modèle informationnel et au modèle de communication de deux organismes mondiaux de normalisation : l'ISO et l'IETF. Le modèle informationnel permet de comprendre comment les composants du système sont perçus par l'administration. C'est à travers une vue logique proposée par chaque composant que l'hétérogénéité du système est occultée. L'analyse du modèle de communication a permis de montrer les moyens d'accès aux différentes vues logique.

Dans la seconde partie nous avons présenté des outils de l'administration disponibles sur le marché.

Le sixième chapitre s'est focalisé sur les plates-formes d'administration et plus particulièrement sur celles des sociétés Bull et Hewlett-Packard. Considérées comme des boîtes à outils, les plates-formes répondent aux diverses activités de l'administration et supportent des développements supplémentaires nécessaires aux besoins spécifiques des clients.

Un exemple d'application spécifique, la distribution de logiciels, a été présenté dans le septième chapitre. La solution de la société Bull à laquelle j'ai participé, appelée SDPC, a été également décrite.

L'objectif principal de ce mémoire était de vous familiariser avec un domaine important de l'informatique : l'administration. Celle-ci répond aux nouveaux besoins organisationnels, à la réduction des coûts, à l'augmentation de la productivité et à l'amélioration du degré de satisfaction de l'utilisateur.

TABLE DES ACRONYMES

A

API	Application Programmer's Interface
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
AVA	Assertion sur Valeur d'Attribut

C

CCITT	Comité Consultatif International pour la Téléphonie et la Télégraphie (ancienne appellation de l'UIT-T)
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CMOT	CMip Over Tcp/ip

D

DCE	Distributed Computing Environnement
DCM	Distributed Computing Model
DEC	Digital Equipment Corporation
DES	Data Encryption Standard
DMI	Définition of Managed Information
DN	Distinguished Name
DoD	Departement of Defense
DSAC	Distributed Systems Administration and Control

E

EGP	Exterior Gateway Protocol
ETSI	European Telecommunications Standard Institute

F

FTP	File Transfert Protocol

FUNDP	Facultés Universitaires Notre-Dame de la Paix (Namur, Belgique)

H

HDLC	High-level Data Link Control
HPOV	Hewlett Packard Open View
HTML	HyperTexte Markup Language

I

IAB	Internet Activities Board
IETF	Internet Engineering Task Force
IHM	Interface Homme-Machine
IP	Internet Protocol
ISM	Integrated System Management
ISO	International Standards Organization
ITU-T	International Telecommunications Union – Telecommunications Sector (synonyme de UIT-T)

L

LPP	Lightweight Presentation Protocol

M

MIB	Management Information Base
MIM	Management Information Model
MIT	Management Information Tree

N

NW/6000	NetView/6000

O

OIM	Osi Internet Management
OMG	Object Management Group
OSF	Open Software Foundation
OSI	Open Systems Interconnection

P

PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data Unit

Q

QoS	Quality of Service

R

RDN	Relative Distinguished Name
RFC	Request For Comment
RNIS-LB	Réseau Numérique à Intégration de Service – Large Bande
RPC	Remote Procedure Call

S

SAP	Service Access Point
SD	Software Distribution
SDPC	Software Distribution for PC
SGBD	Système de Gestion de Base de données
SMF	System Managment Fonction
SMFA	Specific Management Functional Area
SMI	Structure of Management Information
SMTP	Simple Mail Transfer Protocol
SNM	Sun Net Manager
SNMP	Simple Network Management Procotol
SQL	Structured Query Language

T

TCP/IP	Transmission Control Protocol over Internet Protocol
TINA-C	Telecommunications Information Networking Architecture - Consortium

U

UDP/IP	User Datagram Protocol over Internet Protocol
UIT-T	Union Internationale des Télécommunications, secteur Télécommunications (synonyme de ITU-T)

BIBLIOGRAPHIE

[BERN 91] Hervé BERNARD, *"Maîtrise de l'évolution des réseaux"*, article extrait du livre *"De nouvelles architectures pour les communications"*, Eyrolles, Paris, France, 1991.

[BLACK 91] Uyless BLACK, *"OSI, A Model for Computer Standards"*, Prentice-Hall, New Jersey, 1991.

[BRON et al 91] A. BRON, P.A. BOUTROUILLE et J.P. HOUSSART, *"Gestion intégrée et sécurité des systèmes"*, article extrait du livre *"De nouvelles architectures pour les communications"*, Eyrolles, Paris, France, 1991.

[BROWN 97] Article de presse spécialisé, *"Leaders in distributed network management"*, D.H Brown Associates, Octobre 1997.

[BULL IMC 97] Documentation interne de la société BULL, *"Introduction to management concepts"*, BULL, révision 1,1997.

[ELBERT et al 94] Bruce Elbert et Bobby Martyna, *"Client/Server Computing : Architecture, Applications and Distributed Systems Management"*, Artech House, USA, 1994.

[HPOV 98] Guide officiel de HP OpenView : *"HP OpenView Family Guide"*, Hewlett-Packard, mai 1998.

[HUTCH 88] David Hutchison, *"Local Area Network Architectures"*, Addison-Wesley, Great Britain, 1988.

[ISO 10164] ISO 10164-n/ITU-T X.7xy : *"Systems Management Functions"*, 1991.

[ISO 10165-1] ISO 10165-1/ITU-T X.720 : *"Management Information Model"*, 1993.

[ISO 10165-2] ISO 10165-2/ITU-T X.721 : *"Definition of Management Model"*, 1993.

[ISO 10165-4] ISO 10165-4/ITU-T X.722 : *"Guidelines for the Definition of Managed Object"*, 1993.

[ISO 10165-5] ISO 10165-5/ITU-T X.723 : *"Generic Management Model"*, 1994.

[ISO 10165-7] ISO 10165-7/ITU-T X.725 : *"General Relationship Model"*, 1994.

[ISO 9595] ISO/IEC 9595 : *"Common Management Information Service Definition"*, 1991.

[ISO 9596] ISO/IEC 9596 : *"Common Management Information Protocol Definition"*, 1991.

[LABE] J. LABETOULLE, article *"Administration de réseaux, Qualité de service et performances"*, Paris, France.

[LEVA 91] J. LEVASSEUR, *"Protocoles OSI haute performance pour les communications multimédia sur réseaux haut débit"*, article extrait du livre *"De nouvelles architectures pour les communications"*, Eyrolles, Paris, France, 1991.

[LU 96] Guojun Lu, *"Communication and Computing for Distributed Multimedia Systems"*, Artech House, USA, 1996.

[MONTA 98] Jean-Luc MONTAGNIER, *"Pratique des réseaux d'entreprises"*, Eyrolles, Paris, France, Deuxième édition,1998.

[PISC et al 93] David M. PISCITELLO et A. Lyman CHAPIN *"Open Systems Networking, TCP/IP and OSI"*, Addison-Welsey, USA, Août 1993.

[PUJ 95] Guy PUJOLLE, *"Les réseaux"*, chapitre 29 « *La gestion des réseaux* », Eyrolles, Paris, France, 1995.

[RESEAUX-662 96] Revue RESEAUX & TELECOMS, N° 662 01/05/1996, *"Plate-forme d'administration de réseaux : Les caisses de retraite de la SNCF administrées sous Unix"*, 1996.

[RFC –1155] M. ROSE et K. MCCLOGHRIE, *"Structure and Identification of Management Information for TCP/IP-based Internets"*,1990, obsolete the RFC 1065.

[RFC –1157] J. CASE, M. FEDOR, M. SCHOFFSTALL et J. DAVIN, *"A Simple Network Management Protocol (SNMP)"*,1990, obsolete the RFC 1098.

[RFC –1212] M. ROSE et K. MCCLOGHRIE, *"Concise MIB Definitions"*, 1991.

[RFC –1213] M. ROSE et K. MCCLOGHRIE, *"Management Information Base for Network Management of TCP/IP-based Internets : MIB-II"*, 1991, obsolete the RFC 1158.

[RFC –1214] L. LABARRE, *"OSI Internet Management : Management Information Base"*, 1991.

[RFC –1448] J. CASE, K. MCCLOGHRIE, M. ROSE et S. WALDBUSSER, *"Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)"*, 1993.

[ROSE 94] Marshall T. ROSE, un des concepteurs de SNMP, *"The Simple Book : An Introduction to Internet Management"*, Prentice Hall, New Jersey, Deuxième édition,1994.

[ROSEN ET al] Ward ROSENBERRY, David KENNEY et Gerry FISHER, *"Comprendre DCE"*, Addison-Weslay, Paris, France, 1993.

[SDPC 98] Documentation officielle de Bull, *"Software Distribution for PC : Administrator's Guide"*, Bull, Février 1998.

[SIMONI & ZNATY 98] Noëmie SIMONI et Simon ZNATY, *"Gestion de réseau et de service : Similitude des concepts, spécificité des solutions"* , InterEditions, Paris, France, 1998.

[SIMONI et al 91] Noëmie SIMONI et Hidega TIKU, *"Etude de la qualité de service des relais pour l'interconnexion de réseaux haut débit"*, article extrait du livre *"De nouvelles architectures pour les communications"*, Eyrolles, Paris, France, 1991.

[STALL 89] William STALLINGS, *"Handbook of Computer-Communications Standards : The TCP/IP Protocol Suite"*, Howard W. Sams & Company, USA, Volume 3, Seconde édition, 1989.

[STALL 93] William STALLINGS, *"SNMP, $SNMP_{v2}$, and CMIP : The Practical Guide to Network-Management Standards"*, Addison-Wesley, USA, 1993.

[SYLOR 93] Mark SYLOR, *"Junction Objects – A solution to a problem in naming and locating OSI managed objetcs"*, extrait du livre *"Intergrated Network Manangement, III"*, North-Holland, Amsterdam, Hollande, 1993.

[TANEN 92] Andrew TANENBAUM, *"Computer networks"*, Prentice Hall, 1988 (trad. fr. : *"Réseaux – Architecture, protocoles, applications"*, InterEditions, Paris, 3ème tirage corrigé, mai 1992).

[UIT-T M.3100] UIT-T Recommendation M.3100, *"Generic Network Information Model"* , 1993.

[VANB 95] Philippe VAN BASTELAER, *"Description d'un protocole de gestion de réseau : le protocole SNMP"*, extrait du cours de 2ème maîtrise en Informatique (FUNDP), Octobre 1995.

SITES WEB SUR L'INTERNET

- De très bons points d'entrée pour la recherche d'information sur les réseaux et les télécommunications.

  WWW Virtual Library, rubrique Communications & Telecommunications.
  HTTP://WWW.ANALYSYS.CO.UK/COMMSLIB.HTM

  Liste des groupes de News consacrés à l'informatique.
  HTTP://WWW.W3.ORG/HYPERTEXT/DATASOURCES/NEWS/GROUPS/COMP.HTML

  L'Internet Society.
  HTTP://INFO.ISOC.ORG

- Accès aux RFC (*Request For Comment*). Ces documents, souvent issus de discussions informelles, constituent des normes *de facto* dans le domaine des protocoles TCP/IP et des applications Internet. Voici trois façons de se les procurer :

  - soit via un serveur WEB tel que
  HTTP://RFC.FH-KOELN.DE/DOC/RFC/HTML/RFC.HTML

  - soit via un serveur FTP tel que
  FTP://DS.INTERNIC.NET/RFC/RFC-INDEX.TXT

  - soit via le serveur de messagerie
  MAILSERV@DS.INTERNIC.NET

  Par exemple, pour recevoir le RFC 1212 à sa propre adresse de courrier électronique, il suffit d'envoyer à l'adresse mailserv@ds.internic.net un message dont le contenu est : *document-by-name rfc 1212*

- Les organismes de normalisation.

  IEEE (*Institute of Electrical and Electronics Engineers*)
  HTTP://WWW.IEEE.ORG
  FTP://STDSBBS.IEEE.ORG/PUB/802_MAIN

IETF (*Internet Engineering Task Force*)
HTTP://WWW.IETF.ORG
FTP://IETF.ORG/INTERNET-DRAFTS

ISO (*International Standardization Organization*)
HTTP://WWW.ISO.CH

ITU (*International Telecommunications Union*)
HTTP://WWW.ITU.CH

- Articles de presse en ligne.

Réseaux & Télécoms
HTTP://WWW.RESEAUX-TELECOMS.FR

01-Informatique
HTTP://WWW.01-INFORMATIQUE.FR

Data Communications
HTTP://WWW.DATA.COM

Revues de presse
HTTP://WWW.DNA.FR
HTTP://WWW.GEOCITIES.COM

INDEX

A

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T